EXTENDED BAR INDUCTION IN APPLICATIVE THEORIES

Gerard R. RENARDEL DE LAVALETTE
University of Utrecht, Department of Philosophy, Heidelberglaan 2, 3584 CS Utrecht, The Netherlands

Communicated by D. van Dalen
Received 3 January 1990

TAPP is a total applicative theory, conservative over intuitionistic arithmetic. In this paper, we first show that the same holds for TAPP + the choice principle EAC; then we extend TAPP with choice sequences and study the principle EBI₀ (arithmetical extended bar induction of type zero). The resulting theories are used to characterise the arithmetical fragment of EL (elementary intuitionistic analysis) + EBI₀. As a digression, we use TAPP to show that P. Martin-Löf's basic extensional theory ML₀ is conservative over intuitionistic arithmetic.

1. Introduction

1.1. Applicative theories

APP and TAPP are one-sorted intuitionistic theories about a universe of objects, among which the natural numbers and the constants of combinatory logic. These objects can be applied to one another; in APP this application is partial, in TAPP total. We refer to [31, 9.31 for a full definition; a survey of TAPP is in 2.1.

APP is inspired on Feferman's applicative theories ([6], [7]) and is closely related to the theory EON in [2, VI.2], but there is some divergence in the modelling of application. In Feferman's theories, application is partial and compound terms are abbreviations which are explained using the predicate App(x, y, z) with the intended meaning x applied to y yields z: so στ = ρ is inductively defined by

∃xyz (x = σ ∧ y = τ ∧ z = ρ ∧ App(x, y, z)).

In EON and APP, compound terms are no longer abbreviations but real terms that may be undefined, since application is partial; there is an existence predicate and the quantifiers range over existing objects only.

The choice for partial application is straightforward if one considers it as an abstract version of the so-called Kleene-bracket application ⟨·⟩· and continuous function application ·|·, which both are partial operations (on N and N → N, respectively); it is, however, not inevitable. In [18], Klop showed that the partial combinatory algebra 〈N, ⟨·⟩·〉 can be extended to a total one. This insight is
worked out in TAPP, where application is total and every term denotes an
object, so the existence predicate is no longer needed.

TAPP is an extension of APP and hence of HA (intuitionistic arithmetic, also
called Heyting's arithmetic; see [30, 3.3]). To show that it is a conservative
extension of HA, we cannot use the proof for APP by interpreting application by
\{\cdot\}(\cdot); another argument is required, based on arithmetization of a term model of
TAPP (see [31, 9.4.21]).

EAC (extended axiom of choice) is the scheme

\[ \forall x \ (Ax \rightarrow \exists x \ B(x, y)) \rightarrow \exists f \ \forall x \ (Ax \rightarrow B(x, fx)) \quad (\text{no } \lor, \exists \text{ in } A). \]

EAC is equivalent to AC\text{Neg} of [2, VII.3] and can be considered as the abstract
version of ECT\textsubscript{0} (extended Church's thesis, see [30, 4.4.8]).

In 2.2 we shall see that EAC characterizes the realizability interpretation of
TAPP in itself (as ECT\textsubscript{0} does for HA). Together with Skolem functions and
forcing over a monoid, this is used to show that TAPP + EAC is conservative
over HA (Theorem 2.3.14); the argument is in the style of Beeson's proof of
Goodman's theorem (see 2.3.15 for historical remarks). A similar result is proved
in 2.4.10 for extensions of TAPP + EAC with inductive definitions. As an
excursion, we use TAPP to prove in Section 2.5 that P. Martin-Lof's basic
extensional theory ML\textsubscript{0} is conservative over HA; the proof runs parallel to that
methods which yield stronger results but require more difficult proofs).

1.2. Extended bar induction

Bar induction (BI) is the name of a family of closely related axiom schemes of
intuitionistic analysis. One of them reads

\[ \forall \alpha \in \mathbb{N}^\omega \ \exists n \ P(\tilde{a}n) \land \forall xy \ (Px \rightarrow P(x * y)) \]
\[ \land \forall x \ (\forall n \ P(x * \langle n \rangle) \rightarrow Px) \rightarrow P\langle \rangle. \]

Here \( x, y \) range over \( \mathbb{N}^\omega \) (the tree of finite sequences of natural numbers), \( \tilde{a}n \)
abbreviates \( \langle \alpha_0, \ldots, \alpha(n - 1) \rangle \), * denotes concatenation of sequences and \( \langle \rangle \) is
the empty sequence. The first premiss says that \( P \) is a bar in \( \mathbb{N}^\omega \) (it meets every
infinite sequence of nodes), the second that it is monotonic and the third that it is
inductive, so this version of BI can be abbreviated to

\[ \text{Bar}(P) \land \text{Mon}(P) \land \text{Ind}(P) \rightarrow P\langle \rangle. \]

BI is closely related to transfinite induction. Extended bar induction (EBI) is a
generalisation to arbitrary trees \( A \) instead of the fixed tree \( \mathbb{N}^\omega \):

\[ \text{EBI} \quad \text{Tree}(A) \land \text{Bar}(A, P) \land \text{Mon}(A, P) \land \text{Ind}(A, P) \rightarrow P\langle \rangle, \]

where Tree(\( A \)) states that the finite sequences satisfying \( A \) form a tree.

Bar induction is already implicit in L.E.J. Brouwer's writings (e.g. [3]); the first
explicitly formulated version of BI appeared in [17]. The generalization to
extended bar induction is first mentioned in [19]. Several versions have been
studied in the literature: $\text{EBI}_0$, where the $\alpha$ range over some subtree of $\mathbb{N}^{<\omega}$, and $\text{EBI}_1$ with $\alpha$ ranging over some tree of functions of $\mathbb{N} \rightarrow \mathbb{N}$. (For $\text{BI}$, the corresponding versions $\text{BI}_0$ and $\text{BI}_1$ are equivalent modulo continuity assumptions: see [16].) An extensive survey of relevant literature can be found in [28].

In this paper, we distinguish two versions of $\text{EBI}_0$: $\text{EBI}_0^a$, arithmetical $\text{EBI}_0$, where the formula $A$ involved contains no variables for choice sequences, and full $\text{EBI}_0$, where such a restriction is not imposed. It is clear that we have

\[ \text{EBI}_1 \Rightarrow \text{EBI}_0^a \Rightarrow \text{EBI}_0^a \Rightarrow \text{BI}. \]

Now $\text{EL}$ (elementary intuitionistic analysis, see [30, 3.6]) + $\text{EBI}_1$ has the same proof-theoretic strength as full classical analysis [25]. In [20], the extension $\text{CS}$ (choice sequences) of $\text{EL}$ is formulated, in which $\text{BI}$ holds; by an elimination translation, $\text{CS}$ is reduced to $\text{IDB} (= \text{EL} + \text{inductively defined neighbourhood functions})$, a theory with the proof-theoretical strength of $\text{ID}_1 (= \text{HA} + \text{ID}_1$, intuitionistic arithmetic with non-iterated inductive definitions). So $\text{BI}$ is strictly weaker than $\text{EBI}_1$, and the question arises: what is the strength of theories like $\text{EL} + \text{EBI}_0$, $\text{EL} + \text{EBI}_0^a$? In [28], an alleged proof is given of the proof-theoretical equivalence of $\text{EL} + \text{EBI}_0$ and $\text{ID}_{<\omega}$ (intuitionistic arithmetic with finitely iterated inductive definitions); as explained in the Appendix of this paper, a corrected version of this proof only yields proof-theoretical equivalence between $\text{EL} + \text{EBI}_0^a$ and $\text{ID}_1$. This is done there by showing

\[ (*) \quad \text{EL} + \text{EBI}_0^a \text{ is conservative over } \text{ID}_1 \text{ w.r.t. negative arithmetical formulae.} \]

The conjecture that the strength of $\text{EBI}_0$ corresponds to that of $\text{ID}_{<\omega}$ still remains plausible, however (see A.4).

In 3.5.9, we prove a strengthening of $(*)$, viz.

\[ \text{EL} + \text{EBI}_0^a \text{ and } \text{ID}_1 \text{ are arithmetically equivalent,} \]

i.e., they prove the same arithmetical theorems. This is done by shifting from $\text{EL}$ to $\text{TAPP}$, adding choice sequences and inductively defined functionals to $\text{TAPP}$, followed by an elimination translation and some other interpretations which end up in $\text{TAPP(\text{ID})} + \text{EAC}$; this last theory is (by 2.4.10) arithmetically equivalent to $\text{ID}_1$.

The main advantage of $\text{TAPP}$ above $\text{EL}$ in this respect lies in the conservation properties of $\text{TAPP} + \text{EAC}$ and some of its extensions; another point in favour is its type-free character, which admits direct definitions not involving coding. In 3.3, e.g., inductively defined functionals are given directly, not coded by neighbourhood functions as in [20], [28].

1.3. The method of interpretations

This paper can be seen as a study in constructive metamathematics of constructive mathematics, where the last term is taken in the broad sense,
including both intuitionism and constructivism \textit{in sensu stricto}. The metamathematics in this paper is constructive by the proof method used throughout: the method of interpretations. Most other methods used in metamathematics can be classified as either syntactical/proof-theoretic or semantical/model-theoretic. In model theory “everything is allowed”, and the methods used are often not constructive. In proof theory, one reduces arbitrary proofs to proofs in a certain normal form, e.g. by cut elimination or normalization: these methods are usually constructive, and even finitary if the proof systems involved are finite. They work well for logics and weak theories, but stronger theories with axiom schemes for induction often require complicated proof reductions measured by large ordinals.

The method of interpretations is situated in between syntactical and semantical: on the one hand interpretations are defined syntactically, but on the other hand they give an interpretation, i.e., a meaning to terms and formulae. As an example we mention forcing (as treated here in 2.3. and 3.4): it can be considered as a syntactic version of a semantic method, and the formalization (i.e., the transformation into a syntactic translation) is needed here to transform a model construction into a result about formal systems. It is our experience that the flexible character of the theory \textbf{TAPP} makes it particularly apt for this kind of metamathematics.

1.4. Acknowledgements

This paper is a condensed version of Chapters II, III and IV of my thesis [23]. During the research for this thesis I was supported by the Netherlands Organization for the Advancement of Pure Research (then Z.W.O., nowadays N.W.O.). I wish to thank my thesis supervisor A.S. Troelstra for the great help he was to me in the research for and the writing of this paper. He suggested the combination of the proof-theoretic reduction of \textit{EB}_{10} via realizability with Beeson’s notion of forcing [1] used to prove Goodman’s theorem [9].

2. The theory \textbf{TAPP}

2.1. Definition, basic properties

\textbf{APP} is defined in [31, 9.3] as a type-free theory with partial application and an existence predicate \textit{E}. \textbf{TAPP}, the total variant of \textbf{APP}, is obtained by adding the axiom “everything is defined”. Here we consider \textbf{TAPP} as \textbf{APP} without the existence predicate \textit{E}. We now sketch \textbf{TAPP} and some basic properties, referring to [31] for more details.

2.1.1. \textbf{TAPP} has a one-sorted language with the constants \texttt{k}, \texttt{s} (combinators), \texttt{p}, \texttt{p}_0, \texttt{p}_1 (pairing and unpairing), \texttt{0} (zero), \texttt{S} (successor), \texttt{P} (predecessor) and \texttt{d} (definition by cases: \texttt{d}\sigma_1\sigma_2\tau_1\tau_2 is \sigma_1 if \tau_1 and \tau_2 are equal natural numbers, \sigma_2 if
\(\tau_1\) and \(\tau_2\) are different natural numbers). Compound terms are formed by application: if \(\sigma\) and \(\tau\) are terms then so is \(\sigma(\tau)\) (\(\sigma\) applied to \(\tau\), sometimes abbreviated to \(\sigma\tau\)). The only predicate (besides equality) is the unary predicate \(N\); \(N(\tau)\), sometimes written \(\tau \in \mathbb{N}\), means "\(\tau\) is a natural number". Some notation: \(\rho \sigma \tau\) stands for \((\rho \sigma)(\tau)\); we write \(\langle \sigma, \tau \rangle, (\sigma)_0, (\tau)_1\) for \(\rho \sigma \tau, \rho \sigma, \rho \tau; m, n\) are used as numerical variables, so e.g. \(\forall n A\) abbreviates \(\forall n (N(n) \to A)\); \(x, y, n, \ldots\) denote finite sequences of variables \(x_1, \ldots, x_k\), etc.; if we write \(\tau = \tau(x)\) or \(A = A(x)\), we assume that all free variables of \(\tau\) resp. \(A\) are among \(x\).

The connective \(\lor\) is considered to be defined by

\[ A \lor B \equiv \exists x (N(x) \land (x = 0 \to A) \land (\neg x = 0 \to B)). \]

A formula is called arithmetical if all its free and bound variables are restricted to \(\mathbb{N}\); formulae without \(\exists\) (hence also without \(\lor\)) are called negative.

The proof system of \(\text{TAPP}\) consists of first-order intuitionistic logic with equality (as presented, e.g., in [30, Ch. 1]), extended with axioms for the constants and an induction scheme for \(\mathbb{N}\). For some proofs by induction over the length of derivations, we use the following concise axiomatisation of the purely logical part of \(\text{TAPP}\) (a variant of Spector's system [25]; see also [26, 1.1.3]):

\[ \to \text{AX} \quad A \to A \]

\[ \forall \text{AX} \quad \forall x A \to A[x := \tau] \]

\[ \exists \text{AX} \quad A[x := \tau] \to \exists x A \]

\[ \text{PR1} \quad A \Rightarrow B \to A \]

\[ \text{PR2} \quad A \to B, B \to C \Rightarrow A \to C \]

\[ \text{PR3} \quad A, A \to B \Rightarrow B \]

\[ \text{PR4} \quad A \to B, A \to C \Leftrightarrow A \to (B \land C) \]

\[ \text{PR5} \quad (A \land B) \to C \Leftrightarrow A \to (B \to C) \]

\[ \forall \text{R} \quad A \to B \Rightarrow A \to \forall x B \quad (x \text{ not free in } A) \]

\[ \exists \text{R} \quad A \to B \Rightarrow \exists x A \to B \quad (x \text{ not free in } B). \]

2.1.2. Properties. \(\lambda\)-abstraction, a fixpoint operator \(\varphi\) (satisfying \(\varphi xy = x(\varphi x)y\)), a recursor \(R\) (with \(Rxy0=x, Rxy(Sn)=y(Sn)(Rxn)\)) and a minimum operator \(\mu\) (with \(\forall n (fn \in \mathbb{N}) \land \exists n (fn = 0 \land \forall m < n (fm \in \mathbb{N})) \to (\mu f = n \leftrightarrow (fn = 0 \land \forall m < n (fm > 0)))\) are definable in \(\text{TAPP}\) in a standard way; idem for \(\text{APP}\), but some care must be taken in order to keep some subterms defined.

As a consequence, \(\text{HA}\) can be embedded in \(\text{APP}\); on the other hand, \(\text{APP}\) has a direct interpretation into \(\text{HA}\) by interpreting application by Kleene-bracket application. It follows that \(\text{APP}\) is a conservative extension of \(\text{HA}\). (This can all be found in [31, 9.3].) The same fact for \(\text{TAPP}\) requires another argument, based on the formalisation in \(\text{HA}\) of a closed term model for \(\text{TAPP}\) [31, 9.4.21].
2.2. Realizability

Realizability is an interpretation of TAPP into itself, defined by

\[ \tau \ r \ P \overset{\text{def}}{=} P \quad \text{for prime } P \]
\[ \tau \ r \ (A \land B) \overset{\text{def}}{=} ((\tau)_0 \ r \ A) \land ((\tau)_0 \ r \ B) \]
\[ \tau \ r \ (A \land B) \overset{\text{def}}{=} \forall x \ ((x \ r \ A) \rightarrow (\tau x \ r \ B)) \]
\[ \tau \ r \ (\forall x \ A) \overset{\text{def}}{=} \forall x \ (\tau x \ r \ A) \]
\[ \tau \ r \ (\exists x \ A) \overset{\text{def}}{=} (\tau)_1 \ r \ (A[x := (\tau)_0]). \]

Related with realizability is the axiom scheme EAC (extended axiom of choice):

\[ \text{EAC} \quad \forall x \ (Ax \rightarrow \exists y \ B(x, y)) \rightarrow \exists f \ \forall x \ (Ax \rightarrow B(x, fx)) \quad (A \text{ negative}). \]

The restriction to negative \( A \) is essential: otherwise, take \( Ax = \exists y \ (xx \neq y), \)
\( B(x, y) = (xx \neq y), \) then we have \( \exists f \ \forall x \ (\exists y \ (xx \neq y) \rightarrow xx \neq fx) \), so we would get
\( \exists f \ (\exists y \ (ff \neq y) \rightarrow ff \neq ff), \) hence \( \exists f \ \forall y \ \neg \neg (ff = y), \) which is contradictory. As a consequence, TAPP + EAC is essentially non-classical, for in classical logic every formula has a negative equivalent. We have, however:

2.2.1. Lemma. For any formula \( A \) of TAPP there is a negative formula \( A^- \) with

\[ \text{TAPP + EAC} \vdash A \leftrightarrow \exists x \ A^- . \]

Proof. Formula induction, recalling that \( \vee \) is a defined connective and using the equivalences

(i) \( (\exists x \ Ax \land \exists x \ Bx) \leftrightarrow (\exists x \ (A(x)_0 \land B(x)_1)) \)

(ii) \( (\exists x \ Ax \rightarrow \exists x \ Bx) \leftrightarrow (\exists x \ \forall y \ (Ay \rightarrow B(xy))) \quad (A \text{ negative}) \)

(iii) \( \forall y \ \exists x \ A(x, y) \leftrightarrow (\exists x \ \forall y \ A(xy, y)) \)

(iv) \( \exists y \ \exists x \ A(x, y) \leftrightarrow (\exists x \ A((x)_0, (x)_1)). \)

(i), (iv) hold in TAPP, (ii) and (iii) require EAC. \( \square \)

We state some facts about \( r \) and EAC to be used later; the proofs are given in [31, 9.5] (for APP, but they transfer directly to TAPP).

2.2.2. Theorem. (i) \( \text{TAPP} \vdash (\tau \ r \ A)[x := \sigma] \leftrightarrow (\tau[x := \sigma] \ r \ A[x := \sigma]). \)

(ii) \( \text{TAPP + EAC} \vdash A \rightarrow \text{there is a term } \tau \text{ with } \text{TAPP} \vdash \tau \ r \ A. \)

(iii) For negative \( A, \text{TAPP} \vdash A \leftrightarrow \exists x \ (x \ r \ A) \leftrightarrow \forall x \ (x \ r \ A). \)

(iv) \( \text{TAPP + EAC} \vdash A \leftrightarrow \exists x \ (x \ r \ A). \)

(v) \( \text{TAPP + EAC} \vdash A \leftrightarrow \text{TAPP} \vdash \exists x \ (x \ r \ A). \)
2.3. Skolem functions and forcing

We shall prove in this subsection that $\text{TAPP} + \text{EAC}$ is conservative over $\text{HA}$. This is done by the introduction and elimination (by forcing) of Skolem functions for arithmetical formulae $\exists n A(m, n)$, denoted by $\varepsilon_A$. The choice of notation is inspired by Hilbert's $\varepsilon$-symbol (see 2.3.15). We start with defining $\text{TAPP}(\varepsilon)$ by adding the $\varepsilon_A$ to $\text{TAPP}$.

2.3.1. Definition. $\text{TAPP}_\varepsilon$ is $\text{TAPP}$ plus constants $\varepsilon_A$ for every arithmetical formula $A = A(m, n)$ of $\text{TAPP}$ (so $A$ does not contain constants $\varepsilon_B$), and the schema $\varepsilon AX$: this is

$$\varepsilon AX(A) \quad \forall m (\exists n A(m, n) \rightarrow \exists n (A(m, n) \land n = \varepsilon_A m))$$

for all arithmetical $A = A(m, n)$ of $\text{TAPP}$.

Remark. Here $\varepsilon AX$ is equivalent with the somewhat simpler $\forall m (\exists n A(m, n) \rightarrow A(m, \varepsilon_A m))$; this is not the case in the weakening $\text{TAPP}_\varepsilon(A_0)^-$ to be defined in 2.3.7, where quantification over terms containing $\varepsilon$ is restricted.

2.3.2. Definition. The canonical realizer $\tau_A$ of the arithmetical formula $A$ is defined inductively by

$$\begin{align*}
\tau_p & := 0 \quad \text{for prime} \ P \\
\tau_{A \land B} & := (\tau_A, \tau_B) \\
\tau_{A \rightarrow B} & := \lambda n. \tau_B \\
\tau_{\forall n A} & := \lambda n. \tau_A \\
\tau_{\exists n A} & := (\varepsilon_A m, \tau_A[n := \varepsilon_A m]), \quad \text{where} \ m \ \text{are the free variables of} \ \exists n A.
\end{align*}$$

2.3.3. Lemma. For arithmetical $A$ we have

(i) $\text{TAPP}_\varepsilon \vdash A \leftrightarrow \exists x (x \vdash A) \leftrightarrow \tau_A x A$;

(ii) $\text{TAPP} + \text{EAC} \vdash A \Rightarrow \text{TAPP}_\varepsilon \vdash A$.

Proof. (i) Formula induction.

A prime, $A = B \land C$: easy.

$A = B \rightarrow C$: by $\exists x (x \vdash (B \rightarrow C)) \leftrightarrow \exists x \forall y (y \vdash B \rightarrow xy \vdash C) \rightarrow (B \rightarrow \exists z (z \vdash C)) \\
\leftrightarrow (B \rightarrow C) \leftrightarrow (\exists x (x \vdash B) \rightarrow \exists_c t \vdash C) \leftrightarrow k t_c \vdash (B \rightarrow C)$.

$A = \forall n B$: by $\exists x (x \vdash \forall n B) \leftrightarrow \exists x \forall n (x \vdash B) \rightarrow \forall n \exists z (z \vdash B) \leftrightarrow \forall n B \leftrightarrow \\
\forall n (\tau_n x B) \rightarrow \lambda n. \tau_B x \forall n B$.

$A = \exists n B(m, n)$: by $\exists x (x \vdash \exists n B(m, n)) \leftrightarrow \exists x ((x) \vdash B(m, (x) \vdash)) \leftrightarrow \exists y (y \vdash B(m, y)) \leftrightarrow \\
\exists n y (y \vdash B(m, n)) \leftrightarrow \exists n B(m, n) \leftrightarrow B(m, \varepsilon_B m) \leftrightarrow B(m, n)[n := \varepsilon_B m] \\
\leftrightarrow (\tau_B x B(m, n))[n := \varepsilon_B m] \leftrightarrow (\tau_B[n := \varepsilon_B m] \vdash B(m, \varepsilon_B m) \\
\leftrightarrow (\varepsilon_B m, \tau_A[n := \varepsilon_B m]) \vdash \exists n B(m, n)$.

(ii) Follows from (i) and 2.2.2(v). □
2.3.4. With 2.3.3(ii), we are one step away from the desired conservation result: only
\[ \text{TAPP}_\varepsilon \vdash A \Rightarrow \text{TAPP} \vdash A \] for arithmetical A
is required. We shall prove this as follows. If \( \text{TAPP}_\varepsilon \vdash A \), then \( \text{TAPP} + \varepsilon \text{AX}(A_0) + \cdots + \varepsilon \text{AX}(A_k) \vdash A \) for some arithmetical \( A_0, \ldots, A_k \). These \( k+1 \) instances can be reduced to a single one, \( \varepsilon \text{AX}(A_0) \) say, for we have
\[ \text{TAPP} + \varepsilon \text{AX}(A_0) + \cdots + \varepsilon \text{AX}(A_k) \vdash A \Rightarrow \text{TAPP} + \varepsilon \text{AX}(B) \vdash A \]
for \( A \) in the language of \( \text{TAPP} \) and \( B \overset{\text{def}}{=} (n - 0 \land A_0) \lor \cdots \lor (n = k \land A_k) \) \((n \text{ is a fresh variable})\); to see this, take a derivation of \( A \) and replace every occurrence of \( \varepsilon_{A_n} \) by \( \varepsilon_n \). We put
\[ \text{TAPP}_\varepsilon(A_0) \overset{\text{def}}{=} \text{TAPP} + (\text{the constant } \varepsilon) \]
\[ + (\varepsilon \text{AX}(A_0) \text{ with } \varepsilon \text{ instead of } \varepsilon_{A_0}) ; \]
To eliminate \( \varepsilon \) from this theory we define \emph{forcing}, an interpretation in \( \text{TAPP} \). We assume that \( A_0 = A_0(m, n) \).

2.3.5. Definition. Let \( M = M(x) \) be a formula of \( \text{TAPP} \); we write \( \tau \in M \) for \( M(\tau) \). \( M \) is a \emph{monoid} if it contains the identity function and is closed under function composition, i.e.:
\[ \text{TAPP} \vdash \lambda x . x \in M, \]
\[ \text{TAPP} \vdash f, g \in M \rightarrow \lambda x . f(gx) \in M. \]
We use \( f, g, h, k, \ldots \) for elements of \( M \), and define the binary relation \( \geq \) on \( M \) by
\[ f \geq g \overset{\text{def}}{=} f, g \in M \land \exists h \in M \forall x (fx = g(hx)). \]
It is easy to check that \( \geq \) is reflexive and transitive, with \( \lambda x . x \) as minimal element.
\( (M, f) \vDash A \) (\( f \) forces \( A \) in \( M \)) is defined by
\[ (M, f) \vDash P \overset{\text{def}}{=} \forall g \geq f \exists h \geq g \forall k \geq h (P[e := k0]) \text{ for prime } P \]
\[ (M, f) \vDash (A \land B) \overset{\text{def}}{=} (M, f) \vDash A \land (M, f) \vDash B \]
\[ (M, f) \vDash (A \rightarrow B) \overset{\text{def}}{=} \forall g \geq f ((M, g) \vDash A \rightarrow (M, g) \vDash B) \]
\[ (M, f) \vDash (\forall x A) \overset{\text{def}}{=} \forall x ((M, f) \vDash A) \]
\[ (M, f) \vDash (\exists x A) \overset{\text{def}}{=} \forall g \geq f \exists h \geq g \exists x ((M, h) \vDash A). \]
We write \( f \vDash A \) if it is not important which monoid \( M \) is meant, or if this is clear from the context. For \( \forall f \in M ((M, f) \vDash A) \) we write \( M \vDash A \) or \( \vDash A \).

2.3.6. Fact. \( A \Leftrightarrow (M, f) \vDash A \text{ if } A \text{ does not contain } \varepsilon. \)
The thing to do now would be to prove the soundness of $\vdash$ as an interpretation of $\text{TAPP}_e(A_0)$ in $\text{TAPP}$. Unfortunately this is not possible: the monoid $M_0$ we need to get $\varepsilon A X(A_0)$ forced (2.3.10) does not yield, e.g., $M_0 \vDash \exists x (x = \varepsilon)$. The problem lies in quantification over terms containing $\varepsilon$, and forces us to the following detour: we define a weakening $\text{TAPP}_e(A_0)'$ of $\text{TAPP}_e(A_0)$ for which we can prove that $\vdash$ is sound, and we show that $\text{TAPP}_e(A_0)$ can be interpreted in $\text{TAPP}_e(A_0)'$.

2.3.7. Definition. (i) $\text{TAPP}_e(A_0)'$ is $\text{TAPP}_e(A_0)$ with the axiom schemata $\forall A X$, $\exists A X$ restricted to $\tau$ not containing $\varepsilon$ and with the axioms for equality and the constants (except $\varepsilon$) changed into schemes for arbitrary terms (so e.g. $\forall x (x = x)$ becomes the schema $x = x$ for arbitrary terms $x$, possibly containing $\varepsilon$).

(ii) The mapping $^\varepsilon : \text{TAPP}_e(A_0) \rightarrow \text{TAPP}_e(A_0)'$ is defined by

\[
\begin{align*}
x^\varepsilon & := x \varepsilon & (x \text{ a variable}) \\
c^\varepsilon & := c & (c \text{ a constant}) \\
(\sigma \tau)^\varepsilon & := \sigma^\varepsilon \tau^\varepsilon \\
(\sigma = \tau)^\varepsilon & := \sigma^\varepsilon = \tau^\varepsilon \\
(\tau \in \mathbb{N})^\varepsilon & := \exists x \in \mathbb{N} (x = \tau^\varepsilon).
\end{align*}
\]

$^\varepsilon$ commutes with the logical constants

2.3.8. Lemma. (i) $\text{TAPP}_e(A_0)' \vdash \sigma = \tau \rightarrow (A[x := \sigma] \leftrightarrow A[x := \tau])$.

(ii) $\text{TAPP}_e(A_0)' \vdash (\forall x \in \mathbb{N} A)^\varepsilon \leftrightarrow \forall x \in \mathbb{N} (A'[[x] := x])$, $(\exists x \in \mathbb{N} A)^\varepsilon \leftrightarrow \exists x \in \mathbb{N} (A'[x := x])$.

(iii) $\text{TAPP}_e(A_0)' \vdash A^\varepsilon \leftrightarrow A$ for $A$ arithmetical and closed.

(iv) $\text{TAPP}_e(A_0) \vdash A \Rightarrow \text{TAPP}_e(A_0)' \vdash A^\varepsilon$.

Proof. (i) Straightforward. We need the term variant of the equality axioms here, since we no longer have quantification over all terms.

(ii) $(\forall x \in \mathbb{N} A)^\varepsilon = \forall x (\exists y \in n (y = x \varepsilon) \rightarrow A^\varepsilon) \Leftrightarrow \forall x \forall y \in n (y = x \varepsilon \rightarrow A^\varepsilon) \Leftrightarrow \forall x \in \mathbb{N} (A'[[x] := x])$; for the last equivalence we use (i), the fact that $x$ occurs only in the context $x \varepsilon$ in $A^\varepsilon$, and $\forall y \in \mathbb{N} \exists x (y = x \varepsilon)$ (to see this, put $x := ky$). Similarly for the second equivalence.

(iii) With formula induction, using (ii).

(iv) Induction over the length of a derivation of $A$.

Propositional axioms and rules: trivial.

$\forall A X$: by (i) and the definition of $^\varepsilon$ we have $(A[x := \tau])^\varepsilon = A'[[x := \lambda x . \tau]^\varepsilon]$; now $\lambda x . \tau^\varepsilon$ is $\varepsilon$-free, so we have $\forall x A^\varepsilon \rightarrow (A[x := \tau])^\varepsilon$ in $\text{TAPP}_e(A_0)'$, i.e., $(\forall A X)^\varepsilon$.

$\exists A X$: analogously.

$\forall$-R, $\exists$-R: easy.
Axioms for equality and the constants (except $\varepsilon$): follow from the corresponding axioms formulated with terms in $\text{TAPP}_e(A_0)^-$. Induction over $\mathbb{N}$: follows from (ii).
$\varepsilon \text{AX}(A_0)$: follows from (iii). □

Now we look at forcing. For brevity, we put:

$$\text{T}_e \overset{\text{def}}{=} \text{TAPP}(\varepsilon, A_0)^- - \varepsilon \text{AX}(A_0),$$

so $\text{T}_e + \varepsilon \text{AX}(A_0) = \text{TAPP}(\varepsilon, A_0)^-$. We first show that, for every monoid $M$, forcing over $M$ is sound for $\text{T}_e$.

2.3.9. Lemma. $\text{T}_e \vdash A \Rightarrow \text{TAPP} \vdash (\vDash A)$.

Proof. This could be shown directly by induction over the length of a derivation of $A$, but this involves quite a bit of quantifier manipulation. In order to show the structure of the argument more clearly, we choose another way via the auxiliary modal theory $\text{T}_e^{\square}$ and the interpretations $\Box: \text{T}_e \rightarrow \text{T}_e^{\square}$ and $\vDash: \text{T}_e^{\square} \rightarrow \text{TAPP}$.

Definition. $\text{T}_e^{\square} \overset{\text{def}}{=} \text{T}_e +$ the unary connectives $\Box$, $\Diamond$ and the following rule and axiom schemes:

1. $\vdash A \Rightarrow \vdash \Box A$
2. $\Box A \rightarrow A$
3. $\Box A \rightarrow \Box \Box A$
4. $A \rightarrow \Diamond A$
5. $\Diamond \Diamond A \rightarrow \Diamond A$
6. $\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$
7. $\Box (A \rightarrow B) \rightarrow (\Diamond A \rightarrow \Diamond B)$
8. $\forall x \Box A \iff \Box \forall x A$
9. $\Diamond \exists x A \iff \exists x \Diamond A$.

So $\text{T}_e^{\square}$ is the extension of $\text{T}_e$ with the intuitionistic logic $\text{S4}$. We remark that $\Box$ and $\Diamond$ are not interdefinable as in the classical case.

The interpretation $\Box: \text{T}_e \rightarrow \text{T}_e^{\square}$ is defined by

- $P^{\Box} \overset{\text{def}}{=} \Box \Box P$ (P prime)
- $(A \land B)^{\Box} \overset{\text{def}}{=} A^{\Box} \land B^{\Box}$
- $(A \rightarrow B)^{\Box} \overset{\text{def}}{=} \Box (A^{\Box} \rightarrow B^{\Box})$
- $(\forall x A)^{\Box} \overset{\text{def}}{=} \forall x (A^{\Box})$
- $(\exists x A)^{\Box} \overset{\text{def}}{=} \Box \exists x (A^{\Box})$. 
Extended bar induction

\(\Box\)-forcing \(\models\): \(T^\Box_e \rightarrow \text{TAPP}\) is defined by

\((M, f) \models^\Box P \iff P[e := 0]\)

\(\Box^\Box\) commutes with \(\land, \rightarrow, \forall, \exists\)

\((M, f) \models^\Box \Box A \iff \Box g \supseteq f ((M, g) \models^\Box A)\)

\((M, f) \models^\Box \Diamond A \iff \exists g \supseteq f ((M, g) \models^\Box A)\).

We shall show (a)-(c); together they yield the desired result.

(a) \(T^\Box_e \vdash A \Rightarrow \Box^\Box \vdash A^\Box\)

(b) \(T^\Box_e \vdash A \Rightarrow \text{TAPP} \vdash (\Box^\Box \vdash A)\)

(c) \((M, f) \models A \iff (M, f) \models^\Box A^\Box\)

**Proof of (a).** Induction over the length of a derivation of \(A\). Besides (1)–(9), we use the following derivable schemata of \(T^\Box_e\), which follow more or less directly from (1)–(9):

\(\Box (A \land B) \iff (\Box A \land \Box B)\)

\(\Box (A \land B) \rightarrow (\Box A \land \Box B)\)

\(\Box (A \land B) \iff (\Box \Box A \land \Box \Box B)\)

\(\Box (A \rightarrow B) \iff \Box (\Box A \rightarrow \Box B)\)

\(\Box (A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)\)

\(\Box \forall x \vdash \forall x \vdash A\)

\(\exists x \Box A \vdash \Box \exists x A\)

By induction over \(A\) we have, using (1)–(16):

\(\Box^\Box \vdash A^\Box \iff \Box \forall x A^\Box \iff \Box \exists x A^\Box\)

Now we can prove (a). Most rules and axioms go easily: we consider a few cases.

**ζAX:** \((A[x := t] \rightarrow \exists x A)^\Box = \Box (A[x := t]^\Box = \Box \Box \exists x A^\Box = \Box (\Box \Box A[x := t] \rightarrow \Box \Box \exists x A^\Box): for this last step we used (17) and the fact that \(t\) is \(\varepsilon\)-free. The last formula is derivable using (1), (7) and (6).

**PR5:** \((A \land B) \rightarrow C)^\Box = \Box ((A \land B)^\Box \rightarrow C^\Box) \iff \Box (A^\Box \rightarrow (B^\Box \rightarrow C^\Box)) \iff \Box (\Box A^\Box \rightarrow (B^\Box \rightarrow C^\Box)) \iff \Box (\Box (A^\Box \rightarrow (B^\Box \rightarrow C^\Box))) = (A \rightarrow (B \rightarrow C))^\Box), using (17) and (13).

**ζ-R:** \((A \rightarrow B)^\Box = \Box (A^\Box \rightarrow B^\Box) \iff (A^\Box \rightarrow B^\Box) \Rightarrow (\exists x A^\Box \rightarrow B^\Box) \Rightarrow \Box \Box \exists x (A^\Box \rightarrow B^\Box) \Rightarrow \Box \Box \exists x A^\Box \rightarrow B^\Box) \Rightarrow \Box (\Box \Box A^\Box \rightarrow B^\Box) = (\exists x A \rightarrow B)^\Box), using (2), (1), (7), (6) and (17).

Nonlogical axioms (except induction): they all have the form \(P_1 \land \ldots \land P_n \rightarrow Q\) with \(P_1, \ldots, P_n, Q\) prime. Now (assuming \(n = 2\) for simplicity) \((P_1 \land P_2 \rightarrow Q) \Rightarrow \Box \Box \exists (P_1 \land P_2 \rightarrow Q) \Rightarrow \Box (\Box \Box \exists (P_1 \land P_2) \rightarrow \Box \Box \Box Q) \Rightarrow \Box ((\Box \Box P_1 \land \Box \Box P_2) \rightarrow \Box \Box Q) = (P_1 \land P_2 \rightarrow Q)^\Box), using (1), (6), (7) and (12).
Induction over $\mathbb{N}$: \( \text{IND}(A) \equiv \Box(A(0) \land \forall x \Box(\Box \Box N(x) \land A(x) \rightarrow A(Sx)) \rightarrow \forall x \Box(\Box \Box N(x) \rightarrow A(x))) \) and this formula follows from $\Box \text{IND}(A)$, using (3), (6), (10), (8), (7) and (17). This ends the proof of (a).

Proof of (b). Induction over the length of a derivation of $A$. The rules and axioms of $T_e$ are trivial, and (2)–(9) are checked easily. As to (1), we need

\[ \text{TAPP}\vdash ((M,f) \models B) \rightarrow \forall g \geq f ((M,g) \models B) \]

which is proved with induction over $B$. This ends the proof of (b).

Proof of (c). Straightforward. \( \square \)

Now we define the monoid $M_0$ in terms of $A_0$ needed to get $\varepsilon \text{AX}(A_0)$ forced.

2.3.10. Definition.

\[ M_0 \overset{\text{def}}{=} \{ f \mid \forall m \ (\forall x (fxm = xm) \lor \exists n (A_0(m,n) \land \forall x (fxm = n))) \} \]

2.3.11. Lemma. (i) $M_0$ is a monoid. (ii) $\text{TAPP}\vdash (M_0 \models \varepsilon \text{AX}(A_0))$.

Proof. (i) $\lambda x . x \in M_0$ is obvious. Now assume $f, g \in M_0$ and let $m$ be given; by a straightforward case distinction, we get $\forall x (f(gx)m = xm) \lor \exists n (A_0(m,n) \land \forall x (f(gx)m = n))$, and we conclude $f \circ g \in M_0$.

(ii) $M_0 \models \varepsilon \text{AX}(A_0)$ is equivalent to (using that $A_0$ is $e$-free):

\[ \forall m \forall f \ (\exists n A_0(m,n) \rightarrow \\
\quad \forall g \geq f \exists h \geq g \exists n (A_0(m,n) \land \forall y' \geq f' \forall h' \geq g' (n = h'0m)), \]

and this follows from

\[ A_0(m,n_0) \land g \in M_0 \rightarrow \exists h \geq g \exists n (A_0(m,n) \land \forall x (n = hxm)), \]

which we prove as follows. Assume $A_0(m,n_0), g \in M_0$; by the definition of $M_0$, we can distinguish two cases:

(a) $\forall x (gxm = xm)$. Define $h := \lambda xy . \Box n_0(xy)my$, so $hxm = n_0$ and $hxm' = xm'$ if $m' \neq m$; hence $h \in M_0$, $h \geq g$ and $\forall x (hxm = n_0)$, so $\exists h \geq g \exists n (A_0(m,n) \land \forall x (n = hxm))$.

(b) $\exists n (A_0(m,n) \land \forall x (gxm = n))$. Now put $h := g$.

This ends the proof. \( \square \)

2.3.12. Lemma. $\text{TAPP}(\varepsilon, A_0) \vdash A \Rightarrow \text{TAPP} \vdash A$ for arithmetical $A$.

Proof. If $\text{TAPP}(\varepsilon, A_0) \vdash A$, then (by 2.3.8(iii, iv)) $\text{TAPP}(\varepsilon, A_0) \vdash A$, so (by 2.3.9 and 2.3.11) $\text{TAPP} \vdash (M_0 \models A)$ and $\text{TAPP} \vdash A$, by 2.3.6. \( \square \)
With the reasoning in 2.3.4 we now have

**2.3.13. Theorem.** \( \text{TAPP} + \text{EAC} \vdash A \Rightarrow \text{TAPP} \vdash A \) for arithmetical \( A \).

**2.3.14. Corollary.** \( \text{TAPP} + \text{EAC} \) is conservative over \( \text{HA} \).

**Proof.** By 2.1.2 and 2.3.13. \( \Box \)

**2.3.15. Historical remarks.** (i) The idea of Skolem functions first appeared in [24]. Hilbert introduced in [13] the logical function \( \tau(A) \) or \( \tau_\alpha(A(a)) \) with the axiom \( A(\tau(a)) \rightarrow A(a) \); he also mentions the relation with the axiom of choice. In classical logic, \( \tau(A) \) can be thought of as the Skolem function of \( \neg A \); moreover, quantification can be defined with \( \tau \) by

\[
\forall a \ A(a) \equiv A(\tau_\alpha(A(a))), \quad \exists a \ A(a) \equiv \neg A(\tau_\alpha(\neg A(a))).
\]

In [14], Hilbert uses for the first time the symbol \( \varepsilon \) named after him, in the axiom \( A(a) \rightarrow A(\varepsilon A) \).

(ii) In [9], Goodman proves that \( \text{HA}^\omega + \text{AC} \) is conservative over \( \text{HA} \). His proof is based on the interpretation (akin to realizability) of \( \text{HA}^\omega \) into his arithmetic theory of constructions \( \text{ATC} \); in [8] he showed that \( \text{ATC} \) is conservative over \( \text{HA} \) via an argument resembling both forcing and the elimination of choice sequences. He presents a more direct proof in [10] using what he calls relativised realizability, a combination of realizability and forcing. Beeson gives in [1] another proof in which realizability and forcing are used separately. Our proof of \( \text{TAPP} + \text{EAC} \) conservative over \( \text{TAPP} \) is based on a study of Beeson's argument.

**2.4. Inductive definitions**

In this subsection we add non-iterated inductive definitions to \( \text{TAPP} \) and to \( \text{HA} \), and extend the results of the previous subsection. We adopt the following set-and-membership notation, with \( x \) as the designated variable for predicate abstraction:

\[
\tau \in A \ \text{def} \ A[x := \tau] \\
A \subseteq B \ \text{def} \ \forall x \ (x \in A \rightarrow x \in B) \\
A = B \ \text{def} \ A \subseteq B \land B \subseteq A \\
A \cap B \ \text{def} \ A \land B.
\]

**2.4.1. Definition.** (i) The extension \( \text{TAPP(ID)} \) of \( \text{TAPP} \) is defined as follows. First we add unary predicate variables \( P, Q, \ldots \) to the language, with formulae \( \tau \in P \) etc.; call this language \( L_1 \). This language is extended by adding, for every \( L_1 \)-formula \( A \) in which the predicate variable \( P \) occurs only positively (i.e., not in
the left-hand side of an implication), the predicate operator \( \Lambda P \cdot A \) and the unary
predicate \( I(\Lambda P \cdot A) \), leading to formulae \( \tau \in (\Lambda P \cdot A)(B) \) and \( \tau \in I(\Lambda P \cdot A) \). To
the rules and axioms, we add (\( \Gamma \) ranges over predicate operators):

\[
\begin{align*}
P-R & \quad A \Rightarrow A[P := B], \\
\Lambda A X & \quad (\Lambda P \cdot A)(B) \equiv A[P := B], \\
ID1(\Gamma) & \quad \Gamma(I(\Gamma)) \subseteq I(\Gamma), \\
ID2(\Gamma) & \quad \Gamma(P) \subseteq P \rightarrow I(\Gamma) \subseteq P.
\end{align*}
\]

(ii) \( \textbf{ID}_1 \) is defined in the same way, but now starting with \( \textbf{HA} \) instead of \( \textbf{TAPP} \).

2.4.2. \textbf{Fact.} All predicate operators \( \Gamma \) in \( \textbf{TAPP}(\text{ID}) \) are monotonic, i.e., \( P \subseteq Q \rightarrow \Gamma(P) \subseteq \Gamma(Q) \).

2.4.3. \textbf{Definition.} We extend realizability to \( \textbf{TAPP}(\text{ID}) \) by

\[ \tau \mathcal{R} (\sigma \in A) \overset{\text{def}}{=} (\sigma, \tau) \in A', \]

where the mapping \( \mathcal{R} \) is defined by

\[
\begin{align*}
P' & \quad \overset{\text{def}}{=} P \quad \text{for free predicate variables } P \\
\Gamma(A)' & \quad \overset{\text{def}}{=} \Gamma(A') \\
I(\Gamma)' & \quad \overset{\text{def}}{=} f(I(\Gamma)) \\
A' & \quad \overset{\text{def}}{=} (x)_{0} (A[x := (x)_{1}]) \quad \text{for } A \text{ not} \\
& \quad \text{of the form } P, \Gamma(B) \text{ or } I(\Gamma)
\end{align*}
\]

Also

\[
\begin{align*}
\Gamma_{r} & \quad \overset{\text{def}}{=} \Lambda P \cdot (\langle (x)_{0}, \tau(x)_{0}(x)_{1} \rangle \in P), \\
\sigma \cdot \tau & \quad \overset{\text{def}}{=} \lambda x y . \tau x (\sigma x y).
\end{align*}
\]

2.4.4. \textbf{Lemma.} (i) \( \tau \mathcal{R} (A \subseteq B) \leftrightarrow A' \subseteq \Gamma_{r}(B') \).

(ii) \( \Gamma_{\sigma}(\Gamma_{r}(A)) = \Gamma_{\sigma \cdot r}(A) \).

\textbf{Proof.} Straightforward. \( \Box \)

2.4.5. \textbf{Lemma.} \( \textbf{TAPP}(\text{ID}) + \text{EAC} \vdash A \Rightarrow \textbf{TAPP}(\text{ID}) \vdash \tau \mathcal{R} A \) for some term \( \tau \).

\textbf{Proof.} We check the new rule and axioms.

- \( \text{P-R} \): follows from \( \tau \mathcal{R} (A[P := B]) \leftrightarrow (\tau \mathcal{R} A)[P := B'] \), which is proved with induction over \( A \).
- \( \Lambda A X \): follows from \( (A[P := B])[P := B'] \leftrightarrow A'[P := B'] \), proved analogously.
Extended bar induction

ID1(Γ): by ID1(Γ) we have Γ(I(Γ)) ⊆ I(Γ); using Γ, (z ∈ P) = P, we get Γ(I(Γ)) ⊆ Γ, (y(I(Γ))), i.e., ρy . y (I(Γ)) by 2.4.4(i).

ID2(Γ): since Γ is monotonic, there is a σ realizing P ⊆ Q → Γ(P) ⊆ I(Γ(Q)), i.e.

1. ∀u (P ≤ Γn(u) → Γn(u) ≤ Γm(m(Γn(u)))).

We want τ r (Γ(P) ⊆ P → I(Γ) ⊆ P) for some τ, i.e.

2. ∀u (Γn(u) ⊆ Γn(u) → I(Γn(u)) ≤ Γu(Γn(u))).

Assume

3. Γn(P) ≤ Γn(P).

Taking u := τv, P := Γm(P), Q := P in (1), we get

4. Γn(τv) ≤ Γm(τv)(Γn(P)).

(3) implies (using 2.4.4(ii)):

5. Γn(τv)(τv) ≤ Γn(τv)(τv).

Combining (4) and (5):

6. Γn(τv) ≤ Γn(τv)(τv).

Now if τv = σ(τv) · v, then

7. Γn(τv) ≤ Γn(τv),

so by ID2(Γ) we get I(Γ) ≤ Γm(P), the conclusion of (2). So we are ready if τv = σ(τv) · v holds. Here we use the fixed point operator φ mentioned in 2.1.2: put τ := φ(λx . σ(λx . ν) · v), then

τv = φ(λx . σ(λx . ν) · v)v = (λx . σ(λx . ν))v = σ(τv) · v.

Now we turn to forcing. We fix a monoid M and an extension T of TAPP with e-symbols, for which forcing over M is sound. T is extended to T(ID) by adding inductive definitions as in 2.4.1(i).

2.4.6. Definition. Forcing for T(ID) is defined as in 2.3.5, extended with

f ⊩ (τ ∈ A) ≡ (f . τ) ∈ AF,

where the mapping F is defined by

P F def P for free predicate variables P

(A F def I F(A F)

(A F def I F)

(A F def (x) ⊩ (A[x := (x)0]) for A not

of the form P, Γ(B) or I(Γ)

(AP . A) F def AP . A F.
We also put 
\[ [f] \overset{\text{def}}{=} \{ x \mid (x), \geq f \} \).

2.4.7. Lemma. \( f \vdash (A \subseteq B) \iff A^f \subseteq ([f] \to B^f) \).

Proof. Straightforward. \( \Box \)

2.4.8. Lemma. \( \vdash \) is sound for \( T(ID) \).

Proof. We check the new rule and axioms.

P-R: follows from \( (f \vdash A[P := B]) \iff (f \vdash A)[P := B^f] \).

\( \Lambda A X: \) with \( (A[P := B])^f \iff A^f[P := B^f] \).

ID1(\( \Gamma \)): by ID1(\( \Gamma^f \)), we have \( \Gamma^f(\Gamma^f) \subseteq \Gamma^f(\Gamma) \); as \( \Gamma(\Gamma^f) \subseteq ([\lambda x . x] \to \Gamma(\Gamma^f)) \), we get \( \Gamma^f(\Gamma^f) \subseteq ([\lambda x . x] \to \Gamma^f(\Gamma^f)) \), i.e., \( \lambda x . x \vdash (\Gamma(\Gamma^f)) \subseteq \Gamma(\Gamma) \).

ID2(\( \Gamma \)): we want \( \vdash (\Gamma(P) \subseteq P \to \Gamma(P) \subseteq P) \), i.e., for all \( f \in \mathbb{M} \)

\[ (8) \quad \Gamma^f(P) \subseteq ([f] \to P) \to \Gamma^f(P) \subseteq ([f] \to P). \]

So assume \( \Gamma^f(P) \subseteq ([f] \to P) \). This implies

\[ (9) \quad ([f] \to \Gamma^f(P)) \subseteq ([f] \to P). \]

Since \( \Gamma \) is monotonic, we have \( f \vdash Q \subseteq P \to \Gamma(Q) \subseteq \Gamma(P) \), i.e., \( Q \subseteq ([f] \to P) \to \Gamma^f(Q) \subseteq ([f] \to \Gamma^f(P)) \); now take \( Q := ([f] \to P) \), then \( \Gamma^f([f] \to P) \subseteq ([f] \to \Gamma^f(P)) \). With \( (9) \) this yields \( \Gamma^f([f] \to P) \subseteq ([f] \to P) \); combining this with ID2(\( \Gamma^f \)) gives \( \Gamma(\Gamma^f) \subseteq ([f] \to P) \), the conclusion of \( (8) \). \( \Box \)

2.4.9. Theorem. \( TAPP(ID) + EAC \vdash A \Rightarrow TAPP(ID) \vdash A \) for arithmetical \( A \).

Proof. Completely similar to the reasoning in 2.3, using the soundness results 2.4.5 and 2.4.8. \( \Box \)

2.4.10. Corollary. \( TAPP(ID) + EAC \) is conservative over \( ID_1 \).

Proof. Follows from 2.4.9 and a straightforward extension of the proof that \( TAPP \) is conservative over \( HA \) (see 2.2). \( \Box \)

2.5. Extensional types

This subsection is a digression to the basic theory \( ML_0 \) of extensional types by P. Martin-Löf: \( ML_0 \) is the fragment without universes of the theories presented in [21], [22]. As a starting point we take the exposition given in [31, Ch. 11], but drop the type formation operator + (which is definable: see [31, 11.5.5], [29]).

We prove here that \( ML_0 \) is conservative over \( HA \). This is done by defining extensional realizability \( e \) for \( TAPP \), which can be considered as the composition
of the interpretations \( \cdot \) : HA \rightarrow ML_0 \) and \( * : ML_0 \rightarrow TAPP \), extended to the full language of \( TAPP \); the rest of the argument closely follows the argument given in 2.4. Finally we discuss the problem of axiomatising \( \varepsilon \).

The interpretations \( \cdot \) : HA \rightarrow ML_0 \) and \( * : ML_0 \rightarrow TAPP \) are defined by (after [31, 11.4.8, 11.6.4]):

\[
(s = t)^\cdot \overset{\text{def}}{=} I(\mathbb{N}, s^\cdot, t^\cdot)
\]

\[
(A \land B)^\cdot \overset{\text{def}}{=} \Sigma x \in A^\cdot. B^\cdot
\]

\[
(A \rightarrow B)^\cdot \overset{\text{def}}{=} \Pi x \in A^\cdot. B^\cdot
\]

\[
(\forall n A)^\cdot \overset{\text{def}}{=} \Pi n \in \mathbb{N}. A^\cdot
\]

\[
(\exists n A)^\cdot \overset{\text{def}}{=} \Sigma n \in \mathbb{N}. A^\cdot,
\]

\[
n^* \overset{\text{def}}{=} \{(x, x) \mid x \in \mathbb{N}\}
\]

\[
l(A, s, t)^* \overset{\text{def}}{=} \{(0, 0) \mid (s^*, t^*) \in A^*\}
\]

\[
(lx \in A \cdot B)^* \overset{\text{def}}{=} \{(f, g) \mid \forall xy ((x, y) \in A^* \rightarrow (fx, gy) \in B^*)\}
\]

\[
(2x \in A \cdot B)^* \overset{\text{def}}{=} \{(x, y) \mid ((0, 0), (0, 0)) \in A^* \land ((x), (y)) \in B^*[x := (x)_0]\}.
\]

From 11.5.7 (E-HA\(^r\)+ AC \vdash ML_0 \vdash t \in A^* for some \( t \)) and 11.6.5 (if \( ML_0 \vdash \Gamma \vdash \theta \), then \( APP \vdash \Gamma^* \rightarrow \theta^* \)) in [31] it follows that

\[
A \text{ is a sentence with } \text{HA} \vdash A
\]

\[
\implies \text{ there is a closed term with } ML_0 \vdash t \in A^*;
\]

\[
ML_0 \vdash s = t \in A \implies TAPP \vdash (s^*, t^*) \in A^*.
\]

Extensional realizability in \( TAPP \) is defined by

\[
(\sigma, \tau) \in (\rho_1 = \rho_2) \overset{\text{def}}{=} (\sigma = \tau = 0) \land (\rho_1 = \rho_2)
\]

\[
(\sigma, \tau) \in (\rho \in \mathbb{N}) \overset{\text{def}}{=} \rho = \sigma = \tau \in \mathbb{N}
\]

\[
(\sigma, \tau) \in (A \land B) \overset{\text{def}}{=} (((\sigma)_0, (\tau)_0) \in A) \land (((\sigma)_1, (\tau)_1) \in B)
\]

\[
(\sigma, \tau) \in (A \rightarrow B) \overset{\text{def}}{=} \forall xy ((x, y) \in A \rightarrow (\sigma x, \tau y) \in B)
\]

\[
(\sigma, \tau) \in (\forall x A) \overset{\text{def}}{=} \forall x (((\sigma), (\tau)) \in A)
\]

\[
(\sigma, \tau) \in (\exists x A) \overset{\text{def}}{=} \exists x (((\sigma), (\tau)) \in A).
\]

\( \tau \in A \) abbreviates \( (\tau, \tau) \in A \). We have the following properties of \( \varepsilon \):

2.5.1. Lemma. (i) \( \varepsilon \) defines a symmetric relation, i.e., \( (\sigma, \tau) \in A \rightarrow (\tau, \sigma) \in A \).

(ii) \( (\sigma, \tau) \in \forall n A \iff \forall n ((\sigma n, \tau n) \in A) \).

(iii) \( (\sigma, \tau) \in \exists n A \iff (\sigma)_0 = (\tau)_0 \in \mathbb{N} \land ((\sigma)_1, (\tau)_1) \in A[n := (\sigma)_0] \).

(iv) For \( A \) in HA (considered as a subtheory of \( TAPP \)) we have

\[
TAPP \vdash (\sigma, \tau) \in A^* \iff (\sigma, \tau) \in A.
\]

(v) \( TAPP \vdash A \Rightarrow TAPP \vdash \tau \in A \) for some closed term \( \tau \).
Proof. (i)–(iii) Straightforward.

(iv) Induction over \(A\), using (ii) and (iii).

(v) As for \(r\), with some modifications. \(\forall x X\) and \(\exists x X\) are realized by \(\lambda x\ .\ x\); for \(\forall-R\) and \(\exists-R\) we have that the conclusion is \(e\)-realized by the same term as the premiss (using that this term is closed). \(\square\)

For arithmetical \(A\), we define the canonical \(e\)-realizer \(\sigma_A\) in the theory \(\text{TAPP}_e\):

\[
\begin{align*}
\sigma_{p=1} & := 0 \\
\sigma_{p\in \mathbb{N}} & := \rho \\
\sigma_{A \wedge B} & := (\sigma_A, \sigma_B) \\
\sigma_{A \rightarrow B} & := \lambda \sigma_B \\
\sigma_{\forall A} & := \lambda n \ .\ \sigma_A \\
\sigma_{\exists n A} & := (\varepsilon_A m, \sigma_A[n := \varepsilon_A m]).
\end{align*}
\]

Notice the difference with \(\tau_A\), the canonical \(r\)-realizer of \(A\) defined in 2.3.2: \(\sigma_{p\in \mathbb{N}} = \rho\), but \(\tau_{p\in \mathbb{N}} = 0\). Analogously to 2.3.3(i), we have for arithmetical \(A\):

\[
\text{TAPP}_e \vdash A \iff \exists xy ((x, y) e A) \iff \exists x (x e A) \iff \sigma_A e A.
\]

Now we can show that \(\text{ML}_0\) is conservative over \(\text{HA}\). Let \(A\) be an \(\text{HA}\)-sentence which holds in \(\text{ML}_0\), i.e. for which there is a term \(t\) with \(\text{ML}_0 \vdash t \in A^e\). Then \(\text{TAPP} \vdash t^* \in A^e\), so \(\text{TAPP} \vdash t^* e A\), hence \(\text{TAPP}_e \vdash A\); by 2.3.12 and the reasoning in 2.3.4 we have \(\text{TAPP} \vdash A\) and finally \(\text{HA} \vdash A\).

2.5.2. Remarks. (i) It is tempting to think that \((x, y) e A\) is a transitive relation in \(x\) and \(y\); however, the obvious attempt to prove this breaks down at \(A = \exists z B(z)\), for we do not have in general

\[
\exists z ((\rho, \alpha) e R(z)) \land \exists z ((\alpha, \tau) e R(z)) \rightarrow \exists z ((\rho, \tau) e B(z)).
\]

Neither are we able to derive \((\sigma, \tau) e A \rightarrow \sigma e A\). As a consequence, we have no proof of the projectiveness of \(e\): this is the property

\[
\exists xy ((x, y) e A) \iff \exists uv ((u, v) e (\exists xy ((x, y) e A))).
\]

Thus the (obvious) way towards an axiomatisation result for \(e\) is blocked.

Another attempt (which also fails) runs as follows. Define extensional types \(\{0\}, N, S \times T\) and \(S \Rightarrow T\) and formulate the axiom schemes \(\text{EXAC}\) (extensional axiom of choice) and \(\text{EUP}\) (extensional uniformity principle):

\[
\begin{align*}
\text{EXAC} & \quad \forall x \in S (Ax \rightarrow \exists y \in T B(x, y)) \rightarrow \exists f \in (S \Rightarrow T) \forall x \in S (Ax \rightarrow B(x, fx)) \\
& \quad (A \text{ negative}). \\
\text{EUP} & \quad \forall x \exists y \in T A(x, y) \rightarrow \exists y \in T \forall x A(x, y).
\end{align*}
\]

Then \(\text{TAPP} + \text{EXAC} + \text{EUP} \vdash A \iff \exists x (x e A)\) and \(\text{EUP}\) is \(e\)-realizable, but \(\text{EXAC}\) is not.
(ii) Other versions of extensional realizability have been defined and studied in [2, XI.22] and [12]. Our definition differs from those in the treatment of quantification.

3. Extended bar induction

In this section, we study the principle of extended bar induction (EBI, see 1.2). The main result (3.5.8, 3.5.9) will be that \( \text{TAPP}^* + \text{EBI}_0^o \) and \( \text{EL}^* + \text{EBI}_0^a \) both are arithmetically equivalent with \( \text{ID}_1 \). The steps of the argument are:

(i) We define in 3.1 an extension \( \text{TAPP}^* \) of \( \text{TAPP} \) with sequence variables, in which EBI can be formulated properly.

(ii) \( \text{TAPP}^* \) is extended in 3.2 with tree variables, inductively defined sets and choice sequence axioms to \( \text{T}_3^* \), in which EBI is derivable.

(iii) \( \text{T}_3^* \) is in 3.4 interpreted in \( \text{T}_2 \), a theory without sequence variables defined in 3.3.1, by an interpretation that can be considered both as forcing and as an elimination translation in the sense of [20] and [28].

(iv) In 3.5.2, \( \text{T}_2 \) is reduced to \( \text{T}_3 \), a theory without tree variables.

(v) \( \text{T}_3 \) is shown in 3.5.3 to be contained in \( \text{TAPP(ID)} + \text{EAC} \), which is conservative over \( \text{ID}_1 \) (2.4.10).

(vi) Using a result by Sieg [4], we observe in 3.5.6 that \( \text{ID}_1 \) and the subtheory \( \text{ID}_1(\mathcal{O}) \) are arithmetically equivalent.

(vii) Finally we show in 3.5.7 that \( \text{ID}_1(\mathcal{O}) \) is contained in \( \text{TAPP} + \text{EBI}_0^a \), which closes the circle.

3.1. Basic definitions

3.1.1. To formulate EBI we extend \( \text{TAPP} \) to \( \text{TAPP}^* \) by adding new variables \( \alpha, \beta, \ldots \) for sequences of objects; they may occur without restriction in terms and formulae. We add the following quantifier rules and axioms:

\[
\forall R_{\text{SEQ}} \quad \frac{A \rightarrow B}{A \rightarrow \forall \alpha B} \quad (\alpha \text{ not free in } A)
\]

\[
\exists R_{\text{SEQ}} \quad \frac{A \rightarrow R}{\exists \alpha A \rightarrow B} \quad (\alpha \text{ not free in } B)
\]

\[
\forall X_{\text{SEQ}} \quad \forall \alpha \forall n \exists x (\alpha n = x)
\]

\[
\exists X_{\text{SEQ}} \quad \forall \alpha \exists x \forall n (x n = \alpha n)
\]

The other new axioms are:

\[
\text{SEQAX1} \quad \forall \alpha \forall n \exists x (\alpha n = x)
\]

\[
\text{SEQAX2} \quad \forall x \exists \alpha \forall n (x n = \alpha n)
\]

\[
\text{SEQAX3} \quad \forall \alpha \beta \exists \gamma \forall n (\gamma n = (\alpha n, \beta n))
\]

\[
\text{SEQAX4} \quad \forall x \exists \beta (\beta 0 = x \land \forall n (\beta (n + 1) = \alpha n)).
\]
N.B. The axioms $\forall x\ A x \rightarrow A \tau$, $A \tau \rightarrow \exists x\ A x$ remain restricted to $\tau \in \mathcal{L}(\text{TAPP})$; as a consequence, we do not have, e.g., $\exists x\ (x = \alpha)$.

3.1.2. Remark. \text{TAPP}^*$ is the first part of extending \text{TAPP} to $\text{T}_1^*$, a theory with choice sequences (see 3.2). In this sense, \text{TAPP}^* is comparable with \text{EL}^*$ (see [27, 5.2]).

It is consistent to assume $\alpha, \beta, \ldots$ in \text{TAPP}^* to be lawlike (if we consider the objects of \text{TAPP} to be lawlike). This follows from

(1) \text{TAPP}^* \not\vdash \forall \alpha \exists x\ \forall n\ (an = xn),

a consequence of 3.1.4. So the sequences $\alpha, \beta, \ldots$ in \text{TAPP}^* are not really choice sequences yet—that requires CS-like axioms, viz. ECS1–4 in 3.2.1. See also 3.2.6.

3.1.3. The interpretation $A^-$ of a formula $A = A(\alpha, \beta, \ldots)$ of \text{TAPP}^* in \text{TAPP} is straightforward: replace the sequence variables $\alpha, \beta, \ldots$ by object variables $a, b, \ldots$.

3.1.4. Lemma. \text{TAPP}^* \vdash A \Rightarrow \text{TAPP} \vdash A^-.

Proof. Straightforward. \qed

3.1.5. Corollary. \text{TAPP}^* is conservative over \text{TAPP}.

3.1.6. The sequences $\alpha, \beta, \ldots$ we introduced above can be looked at from two points of view:

(i) as objects (not in the range of the variables $x, y, \ldots$ of \text{TAPP}) with some special properties as stated in the axioms: the corresponding equality is $\alpha = \beta$, equality between objects;

(ii) as sequences of objects $\alpha 0, \alpha 1, \ldots$: here the appropriate equality is $\alpha \equiv \beta$, where $\equiv$ is defined by

$$(\alpha = \tau) : = \forall n\ (an = \tau n).$$

Warning. The rôle of $=, \equiv$ is not the same as in other publications on choice sequences.

Now it is the second point of view which concerns us here, and we would like to have the following substitution property:

(2) $\alpha = \beta \rightarrow (A\alpha \leftrightarrow A\beta)$.

(2) is derivable in \text{TAPP}^* in case $\alpha$ occurs regularly in $A\alpha$, i.e., only in contexts $\alpha \tau$ where $\tau$ is a natural number.
3.1.7. **Definition.** (i) A formula $A$ of **TAPP**$^*$ is called *regular* if all its free
sequence variables occur regularly in $A$.

(ii) A formula $A$ is called *totally regular* if all its (free and bound) sequence
variables occur regularly in $A$.

We do not want to restrict our formal language to regular formulae in order to
obtain (2): that would require a complicated definition of different sorts of terms,
conflicting with the type-free and flexible character of **TAPP**. To be able to
formulate a weaker but valid version of (2), we use a well-known method for
making predicates extensional: define

$$A(\alpha_1, \ldots, \alpha_n)^e := \exists \beta_1 \cdots \beta_n (\alpha_1 = \beta_1 \land \cdots \land \alpha_n = \beta_n \land A(\beta_1, \ldots, \beta_n));$$

here $\alpha_1, \ldots, \alpha_n$ are the sequence variables occurring free in $A$. Now $A^e$ is always
regular and we have, in **TAPP**$^*$:

$$A \rightarrow A^e,$$

$$A \leftrightarrow A^e \quad \text{for regular } A,$$

$$\alpha = \beta \rightarrow (A\alpha)^e \leftrightarrow (A\beta)^e.$$  

3.1.8. **Some notation and conventions.** A finite sequence $x_0, \ldots, x_{n-1}$ is coded
by an object $f$ iff:

\[
\begin{cases}
  f^0 = n, \\
  f(i + 1) = x_i & (0 \leq i < n).
\end{cases}
\]

This coding is not unique, of course: one easily constructs $f, g$ with $f^0 = g^0 = n,$
$\forall i < n (f(i + 1) = g(i + 1))$ and $f(n + 1) \neq g(n + 1)$. However, we shall write
$\langle x_0, \ldots, x_{n-1} \rangle$ for $f$ satisfying (3), but only in cases where no ambiguity can
occur.

It is not hard to define in **TAPP** the functions $\langle \cdot \rangle, \text{lth}, *, \cdot, ^-, (\cdot)$ satisfying

$$\langle x_0, \ldots, x_{n-1} \rangle_i = x_i \quad (0 \leq i < n)$$

$$\text{lth}(\langle x_0, \ldots, x_{n-1} \rangle) = n$$

$$\langle x_0, \ldots, x_{n-1} \rangle \cdot \langle y_0, \ldots, y_{m-1} \rangle = \langle x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1} \rangle$$

$$\hat{x} = \langle x \rangle$$

$$\text{lth}(\langle \rangle) = 0$$

$$\hat{a_n} = \langle a_0, \ldots, a(n - 1) \rangle.$$  

The equivalence relation $\sim$ between finite sequences is defined by

$$x \sim y := (\text{lth } x = \text{lth } y \in \mathbb{N} \land \forall i < \text{lth } x ((x)_i = (y)_i)).$$

$*$ is also used to denote concatenation of a finite sequence with an infinite one: if
a is (thought of as) an infinite sequence $a_0, a_1, \ldots$, then

\[
((x_0, \ldots, x_{n-1}) \ast a)m = \begin{cases} x_m & \text{if } m < n, \\ a(n - m) & \text{if } m \geq n. \end{cases}
\]

In the sequel, we shall often use the notation $\phi_x$, defined by

\[\phi_x := \lambda x. \phi(x \ast a).\]

3.1.9. The set-and-membership notation of 2.4 is extended with

\[A \Rightarrow B := A \rightarrow B.\]

We also put

\[
\begin{align*}
\tau \in x & := \text{lth } x \in \mathbb{N} \land \forall i < \text{lth } x \ (\tau_i = (x)_i) \\
\tau \in \bar{A} & := \forall n \ (\tau n \in A) \\
\tau \in A_x & := x \ast \tau \in A \\
\mathbb{N}^{<\omega} & := \text{lth } x \in \mathbb{N} \land \forall i < \text{lth } x \ ((x)_i \in \mathbb{N}) \\
\mathbb{N}^{\omega} & := \forall n \ (\tau n \in \mathbb{N}) \\
\text{Tree}(A) & := \forall x \in A \ (\text{lth } x \in \mathbb{N}) \\
& \quad \land \forall x y (x \sim y \land x \ast y \in A \rightarrow y \in A) \land \langle \rangle \in A \\
& \quad \land \forall x y (x \ast y \in A \rightarrow x \in A) \land \forall x \in A \exists y (x \ast y \in A).
\end{align*}
\]

3.1.10. To the equivalence relations $\sim$ (for finite sequences) and $\equiv$ (for sets), we add:

\[
\begin{align*}
\sigma = \tau & := \forall n \ (\sigma n = \tau n), \\
\phi =_A \psi & := \forall x \in \bar{A} \ (\phi x = \psi x) \\
f =_A g & := \forall x \in \bar{A} \ (f x = g x).
\end{align*}
\]

These relations satisfy the following properties.

3.1.11. Lemma. (i) $x \sim y \land x \in A \rightarrow \alpha \in y$.
(ii) $x \sim y \land x \in A \land \text{Tree}(A) \rightarrow y \in A$.
(iii) $x \sim y \rightarrow x \ast \alpha = y \ast \alpha$.
(iv) $x \sim y \land \text{Tree}(A) \rightarrow A_x = A_y$.
(v) $\alpha = \beta \rightarrow \alpha n = \beta n$.
(vi) $\alpha = \beta \land \alpha \in x \rightarrow \beta \in x$.
(vii) $\phi =_A \psi \land x \in A \land \text{Tree}(A) \rightarrow \phi_x =_A \psi_x$. 
(viii) \( A = B \rightarrow \bar{A} = \bar{B} \).
(ix) \( A = B \land \text{Tree}(A) \rightarrow \text{Tree}(B) \).
(x) \( \text{Tree}(A) \rightarrow A = A \).

**Proof.** Easy. \( \square \)

### 3.1.12. Definition of EBl\(_a\)

We define

\[
\begin{align*}
\text{Bar}(A, P) & := \forall x \in A \exists n P(\bar{a}n), \\
\text{Mon}(A, P) & := \forall xy (x \ast y \in A \land Px \rightarrow P(x \ast y)), \\
\text{Ind}(A, P) & := \forall x \in A (\forall y (x \ast y \in A \rightarrow P(x \ast y)) \rightarrow Px).
\end{align*}
\]

Now EBl\(_0\)(A, P) reads

\[
\text{Tree}(A) \land \text{Bar}(A, P) \land \text{Mon}(A, P) \land \text{Ind}(A, P) \rightarrow P(\).
\]

EBl\(_0\)(A) is EBl\(_0\)(A, P) for all regular \( P \in \mathcal{L}(\text{TAPP}^*) \), and EBl\(_0^\circ\) is EBl\(_0\)(A) for all \( A \in \mathcal{L}(\text{TAPP}) \) (hence not containing sequence variables). BI is defined as EBl\(_0\)(N\( \omega^\omega \)).

### 3.2. An applicative theory with choice sequences

In this section we define the theory \( T^*_1 \) and show, among other things, that \( T^*_1 \vdash \text{EBI}^0_0 \).

### 3.2.1. The language of \( T^*_1 \)

The language of \( T^*_1 \) consists of that of \( \text{TAPP}^* \) plus variables \( S, T, \ldots \) for trees and the constants \( U \) (the universal tree) and \( I_0 \) (for inductively defined sets of functions). \( T^*_1 \) has tree terms, defined as follows:

(i) \( U \) and all tree variables are tree terms.
(ii) If \( V, W \) are tree terms, then so is \( V \times W \).
(iii) If \( V \) is a tree term and \( \tau \) a term, then \( V_{\tau} \) is a tree term.

New prime formulae are \( \tau \in V \) and \( \tau \in I_0(V) \), \( V \) a tree term. We assume \( \times, [ ]_0, [ ]_1 \) to be defined satisfying

\[
\begin{align*}
\langle x_0, \ldots, x_{n-1} \rangle \times \langle y_0, \ldots, y_{n-1} \rangle &= \langle \langle x_0, y_0 \rangle, \ldots, \langle x_{n-1}, y_{n-1} \rangle \rangle, \\
[\langle x_0, \ldots, x_{n-1} \rangle]_i &= \langle \langle x_0 \rangle_i, \ldots, \langle x_{n-1} \rangle_i \rangle \quad (i = 0, 1).
\end{align*}
\]

Now we can give the new rules and axioms of \( T^*_1 \):

\[
\begin{align*}
\forall \text{R}_\text{TR} & \quad \frac{A \rightarrow B}{A \rightarrow \forall T B} \quad (T \text{ not free in } A) \\
\exists \text{R}_\text{TR} & \quad \frac{A \rightarrow B}{\exists T A \rightarrow B} \quad (T \text{ not free in } B) \\
\forall \text{A}_\text{TR} & \quad \forall T A(T) \rightarrow A(S) \\
\exists \text{A}_\text{TR} & \quad A(S) \rightarrow \exists T A(T)
\end{align*}
\]
TRAX1 Tree(T) for all tree variables T

TRAX2 \( \tau \in U \iff \text{1th } \tau \in \mathbb{N} \)

TRAX3 \( \sigma \in V, \leftrightarrow \tau \ast \sigma \in V \)

TRAX4 \( \tau \in V \times W \iff [\tau]_0 \in V \land [\tau]_1 \in W \)

TRAX5 Tree(A) \( \rightarrow \exists T \ (T = A) \) for \( A \in \mathcal{L}^{-}(\text{TAPP}) \)

\( \text{(i.e. } A \in \mathcal{L}(\text{TAPP}), A \text{ negative) } \)

TRAX6 \( \forall T \forall x (x \in T \rightarrow \exists S (S = T_x)) \)

TRAX7 \( \forall TT' \exists S (S = T \times T') \)

TRAX8 \( I_0(T) = I_0(T_1) \).

In \( I_0AX1-3, I_0AX \) we use \( \phi, f \) as variables of \( \text{TAPP} \) (i.e. ranging over objects). In the rest of this chapter, we shall often use \( \phi \) and \( \psi \) for elements of some \( I_0(T) \), and \( f, g, h, \ldots \) for elements of some \( I_i(S, T) \).

\( I_0AX1 \quad \forall \alpha \in \bar{T} \ (\phi \alpha = x) \rightarrow \phi \in I_0(T) \)

\( I_0AX2 \quad \forall \alpha \in T \ (\phi \alpha \in I_0(T_1)) \rightarrow \phi \in I_0(T) \)

\( I_0AX3 \quad \forall \alpha \in T \forall \phi \exists \psi \forall \alpha \in \bar{T} \ (\phi \alpha = y) \rightarrow \forall \psi \in T_x (\phi_x \in P(x \ast y)) \rightarrow \phi \in P_x \rightarrow \forall \psi \in T \ (I_0(T_1) \subseteq P(x)) \)

\( I_1AX \quad \forall \alpha \in S \forall f \in I_1(S, T) \exists \beta \in T \ (\beta = f \alpha) \).

Here \( I_i \) is defined by

\[ f \in I_i(S, T) \iff \forall n \ (\lambda x. f x n \in I_0(S)) \land \forall a \in S \ (fa \in \bar{T}). \]

In the next five axioms, \( A \) and \( B \) contain no free sequence variables besides those shown.

ECS1 \( \forall \alpha \in \bar{T} \ A \alpha \rightarrow \forall \alpha \in \bar{T} A \alpha \quad \text{for prime} \ A \)

ECS2 \( \forall T \forall f \in I_1(T, U) (\forall \alpha \in \bar{T} A(f \alpha) \rightarrow \forall \alpha \in \bar{T} B(f \alpha)) \rightarrow \forall \alpha (A \alpha \rightarrow B \alpha) \)

ECS3 \( \forall \alpha \in \bar{T} \exists x A(\alpha, x) \rightarrow \exists \phi \in I_0(T) \forall \alpha \in \bar{T} A(\alpha, \phi \alpha) \)

ECS4 \( \forall \alpha \in \bar{T} \exists \beta A(\alpha, \beta) \rightarrow \exists f \in I_1(T, U) \forall \alpha \in \bar{T} A(\alpha, f \alpha) \)

EAC \( \forall x (Ax \rightarrow \exists y B(x, y)) \rightarrow \exists f \forall x (Ax \rightarrow B(x, fx)) \quad A \text{ negative.} \)

3.2.2. Remarks. (A) Not all tree terms \( V \) satisfy Tree(\( V \)) : e.g., for \( V = T \), this is only the case if \( \tau \in T \).

(B) By \( I_0AX1-3, I_0(T) \) is an inductively defined set of functions \( \phi \) defined on sequences \( \alpha \) with \( \forall n (\bar{\alpha} n \in T) \) (so \( \alpha \) is an "infinite branch" of \( T \)). \( I_0AX1 \) states that all constant functions \( \phi \) are in \( I_0(T) \), by \( I_0AX2 \) one can prove, e.g., that \( \lambda \alpha. \alpha 0, \lambda \alpha. \alpha 1, \ldots \) are in \( I_0(T) \); the schema \( I_0AX3 \) expresses that \( I_0(T) \) is the smallest set satisfying \( I_0AX1 \) and \( I_0AX2 \).
$I_1(S, T)$ is a set of functions from $S$ to $T$, and consists by definition of those functions the projections of which are elements of $I_0(S)$.

$I_0(T)$, $I_1(S, T)$ are sometimes called $I_o$ resp. $I_1$-sets. They are investigated in 3.3.

(C) Comparing $T^*_1$ with $CS^*$ in [28], we observe the following differences (besides the choice of APP resp. EL as basic system):

(i) $T^*_1$ has tree variables, whereas $CS^*$ has type constants (types are subsets of $\mathbb{N}$).

It is shown in [28, 2.5, 2.6] that $EBI_0(A)$ ($EBI^*(A)$ in the notation used there), $A$ a subtree of $\mathbb{N}^{<\omega}$, can be reduced to $EBI_0(B^{<\omega})$, $B \subseteq \mathbb{N}$; however, this method of reduction is based on decidable equality on $\mathbb{N}$, and can therefore not be applied in our context (unless we would restrict $EBI^*_0$ to subtrees of $\mathbb{N}^{<\omega}$).

Tree variables in $T^*_1$ are needed to formulate the axiom ECS2; it is weaker than its counterpart in $CS^*$.

(ii) The functionals in $I_0(T)$, $I_1(S, T)$ are not coded by neighbourhood functions as in $CS^*$ (using $K_\sigma$, $K_\alpha$), but are directly present in $T^*_1$; this allows a more direct treatment (cf. 3.3).

(iii) The trees in $T^*_1$ for which $I_0(S)$ is defined can be seen as trees definable in TAPP; hence $I_0\text{AX}1-3$ may be thought of as a schema of non-iterated inductive definitions. In $CS^*$, however, the defining formula of a type $\sigma$ may contain inductively defined sets $K_\alpha$, which makes the defining axioms of the $K_\sigma$ equivalent to finitely iterated inductive definitions.

3.2.3. We now give some properties of $T^*_1$. In some proofs, we use facts about $I_0$, $I_1$ which are proved afterwards in 3.3.

3.2.4. Lemma. $\forall T \exists a (a \in \hat{T})$.

Proof. $\text{Trcc}(T)$, so $\forall x \in T \exists y (x \ast y \in T)$. With $EAC$: $\exists f \forall x \in T (x \ast (fx) \in T)$. Now define

$$a0 := f(\langle \rangle), \quad a(n + 1) := f(an),$$

then $\forall n (\forall n \in T)$. □

3.2.5. Corollary. $\forall T \exists x (x \in \hat{T})$ (by $SEO\text{AX}2$).

We show that $T^*_1$ is a proper extension of $TAPP^*$.

3.2.6. Lemma. $T^*_1 \vdash \forall x \forall n (x(n) = an)$.

Proof. Assume $\forall x \forall n (x(n) = an)$, then (by ECS3) $\forall x \forall n (\phi \cup n = an)$ for some $\phi \in I_0(U)$. But by 3.3.10.(i) such a $\phi$ is continuous, so the value of $\phi \alpha$ is determined by an initial segment of $\alpha$: contradiction. □
3.2.7. **Corollary.** \( T^*_t \) properly extends \( T\text{APP}^* \).

**Proof.** Combine 3.2.6 with (1) in 3.1.2. \( \Box \)

3.2.8. **Definition.** We define four schemata: EAD, ECS2', ECS3' and ECS4'.

EAD is a weakening of the axiom of analytic data AD in [28]; ECS2' is a relativized version of ECS2; ECS3' and ECS4' are extensions of ECS3 and ECS4 to arbitrary regular formulae \( A \).

\[
\text{EAD} \quad A\alpha \rightarrow \exists T \exists f \in I_1(T, U)(\exists \beta \in \tilde{T} \ (f\beta = \alpha) \land \forall \beta \in \tilde{T} A(f\beta))
\]

\[
\text{ECS2'} \quad \forall S \forall f \in I_1(S, T)(\forall \alpha \in \tilde{S} A(f\alpha) \rightarrow \forall \alpha \in \tilde{S} B(f\alpha)) \rightarrow \forall \alpha \in \tilde{T} \ (A\alpha \rightarrow B\alpha)
\]

In EAD, ECS2', \( A \) contains no free sequence variables besides \( \alpha \).

\[
\text{ECS3'} \quad \forall \alpha \in \tilde{T} \exists x A(\alpha, x) \rightarrow \exists S \exists \gamma \in \tilde{S} \exists \phi \in I_0(S \times T) \forall \alpha \in \tilde{T} A(\alpha, \phi(\gamma \times \alpha))
\]

\[
\text{ECS4'} \quad \forall \alpha \in \tilde{T} \exists \beta A(\alpha, \beta) \rightarrow \exists S \exists \gamma \in \tilde{S} \exists f \in I_1(S \times T, U) \forall \alpha \in \tilde{T} A(\alpha, f(\gamma \times \alpha))
\]

In ECS3', ECS4', \( A \) is regular and may contain free sequence variables besides \( \alpha \).

3.2.9. **Lemma.** (i) EAD and ECS2 are equivalent, i.e.

\[ T^*_t \vdash \text{ECS2} \iff \text{EAD} \]

(ii) \( T^*_t \vdash \text{ECS2}'. \)

**Proof.** (i) \((\rightarrow)\) By logic, we have

\[
\forall S \forall g \in I_1(S, U)(\forall \alpha \in \tilde{S} A(g\alpha)) \rightarrow \forall \alpha \in \tilde{S} \exists T \exists f \in I_1(T, U)(\exists \beta \in \tilde{T} \ (f\beta = g\alpha) \land \forall \beta \in \tilde{T} A(f\beta))
\]

to see this, take \( T := S \), \( f := g \). So, by ECS2 we have

\[
\forall \alpha \ (A\alpha \rightarrow \exists T \exists f \in I_1(T, U)(\exists \beta \in \tilde{T} (f\beta = \alpha) \land \forall \beta \in \tilde{T} A(f\beta)))
\]

i.e. EAD.

\((\leftarrow)\) Assume

(1) \( \forall T \forall f \in I_1(T, U)(\forall \alpha \in \tilde{T} A(f\alpha) \rightarrow \forall \alpha \in \tilde{T} B(f\alpha)) \)

take any \( \alpha \) and assume \( A\alpha \). By EAD:

\[
\exists S \exists g \in I_1(S, U)(\exists \beta \in \tilde{S} (g\beta = \alpha) \land \forall \beta \in \tilde{S} A(g\beta))
\]

so, by (1)

\[
\exists S \exists g \in I_1(S, U)(\exists \beta \in \tilde{S} (g\beta = \alpha) \land \forall \beta \in \tilde{S} A(g\beta))
\]

and hence \( B\alpha \), by the substitution property of \( = \).

(ii) Easy, take \( \alpha \in \tilde{T} \land A(\alpha) \) for \( A \). \( \Box \)
3.2.10. Lemma (i) For regular $A$, we have in $T_1^*$

\[(2) \forall \alpha \in \bar{S} \forall \beta \in \bar{T} A(\alpha, \beta) \iff \forall \alpha \in \bar{S \times T} A(\pi_0 \alpha, \pi_1 \alpha)\]

(see 3.3.6 for a definition of $\pi_0$, $\pi_1$).

(ii) $T_1^* \vdash$ ECS3', ECS4'.

Proof. (i) By SEQAX3, we have $\forall \alpha \in \bar{S} \forall \beta \in \bar{T} \exists \gamma \in \bar{S \times T} (\gamma = \alpha \times \beta)$ ($\alpha \times \beta$ is defined in 3.3.6) and, by 3.3.8.(v) and $I_1AX$, we also have $\forall \gamma \in \bar{S \times T} \exists \alpha \in \bar{S} \exists \beta \in \bar{T} (\pi_0 \gamma = \alpha \land \pi_1 \gamma = \beta)$. Together with the substitution property for $= w.r.t. regular formulae (3.1.7) this yields (2).

(ii) We first prove ECS3'. Assume

\[\forall \alpha \in \bar{T} \exists x A(\alpha, x)\]

where $A$ is regular. Without loss of generality we assume that $A$ contains as free sequence variables besides $\alpha$ only $\beta_0$ and $\beta_1$, so $A = A(\alpha, x, \beta_0, \beta_1)$. Let $\beta := \beta_0 \times \beta_1$. Then, by (i)

\[\forall \alpha \in \bar{T} \exists x A(\alpha, x, \pi_0 \beta, \pi_1 \beta).\]

By EAD (which is derivable in $T_1^*$, by 3.2.9.(i)), there are $S$, $f \in I_1(S, U)$, $\gamma_0 \in \bar{S}$ with $f \gamma_0 = \beta$ and

\[\forall \gamma \in \bar{S} \forall \alpha \in \bar{T} \exists x A(\alpha, x, \pi_0 (f \gamma), \pi_1 (f \gamma)).\]

Apply (i):

\[\forall \alpha \in \bar{S \times T} \exists x A(\pi_0 \alpha, x, \pi_0 (f (\pi_1 \alpha)), \pi_1 (f (\pi_1 \alpha))).\]

Now with ECS3:

\[\exists \phi \in I_0(S \times T) \forall \alpha \in \bar{S \times T} A(\pi_0 \alpha, \phi \alpha, \pi_0 (f (\pi_1 \alpha)), \pi_1 (f (\pi_1 \alpha)))\]

which is equivalent to

\[\exists \phi \in I_0(S \times T) \forall \gamma \in \bar{S} \forall \alpha \in \bar{T} A(\alpha, \phi (\gamma \times \alpha), \pi_0 (f \gamma), \pi_1 (f \gamma))\]

hence (take $\gamma := \gamma_0$, and use $f \gamma_0 = \beta$, $\beta = \beta_0 \times \beta_1$ and (i))

\[\exists \phi \in I_0(S \times T) \forall \alpha \in \bar{T} A(\alpha, \phi (\gamma_0 \times \alpha), \beta_0, \beta_1)\]

so

\[\exists S \exists \gamma \in \bar{S} \exists \phi \in I_0(S \times T) \forall \alpha \in \bar{T} A(\alpha, \phi (\gamma \times \alpha), \beta_0, \beta_1).\]

ECS4' is derived analogously. \qed

3.2.11. Definition. EIUS, extended induction over unsecured sequences, is defined by

\[EIUS \quad \forall S \forall \gamma \in \bar{S} \forall \phi \in I_0(S \times T) \land (\bar{S \times T} \Rightarrow \mathbb{N})\]

\[(\forall \alpha \in T Q(\bar{x}(\phi (\gamma \times \alpha))) \land \text{Mon}(T, Q) \land \text{Ind}(T, Q) \rightarrow Q(\bar{x})).\]

3.2.12. Lemma. $T_1^* \vdash$ EIUS.
Proof. Use $\text{L}_0\text{AX}3$ with
\[
\phi \in P(x) := \phi \in ((S \times T)_x \Rightarrow \mathbb{N}) \land \forall \alpha \in T, Q(x \ast a(\phi(\gamma \times \alpha))) \\
\land \text{Mon}(T, Q) \land \text{Ind}(T, Q)
\]
to prove
\[
\forall x \in T \forall \gamma \in S \forall \phi \in \text{L}_0((S \times T)_x) \cap ((S \times T)_x \Rightarrow \mathbb{N}) \land \forall \alpha \in T, Q(x \ast a(\phi(\gamma \times \alpha))) \land \text{Mon}(T, Q) \land \text{Ind}(T, Q) \Rightarrow Q(x);
\]
then take $x := (\ )$. For details, see 3.2.1 and 5.7.4 in [20]. □

3.2.13. Lemma. $T^*_1 \vdash \text{EBI}_0(A)$ for $A \in \mathcal{L}^-(\text{TAPP})$.

Proof. Assume Tree($A$), then $A = T$ for some $T$ (by TRAX5); and by ECS3'
\[
\forall \alpha \in T, \exists n P(\bar{\alpha}n) \Rightarrow \exists S \exists \gamma \in S \exists \phi \in \text{L}_0(S \times T) \cap (S \times T \Rightarrow \mathbb{N}) \land \forall \alpha \in T, Q(\bar{\alpha}(\phi(\gamma \times \alpha)))
\]
for regular $P$. Now apply EIUS. □

3.2.14. Theorem. $T^*_1 \vdash \text{EBI}_0(A)$ for all $A \in \mathcal{L}(\text{TAPP})$.

Proof. Let $A \in \mathcal{L}(\text{TAPP})$. By 2.2.1 we have $x \in A \Leftrightarrow \exists y A^-(x, y)$ for some $A^- \in \mathcal{L}^-(\text{TAPP})$. Assume Tree($A$), Bar($A, P$), Mon($A, P$), Ind($A, P$), and define
\[
x^k := \lambda n. (x)_n^k,
\]
so $\langle x_0, \ldots, x_{n-1} \rangle^k = (x_0)_0, (x_1)_0, \ldots, (x_{k-1})_0$, and put
\[
x \in B := \text{lth } x \in \mathbb{N} \land \forall n < \text{lth } x A^-(x^n, ((x)_n)), \\
Q(x) := P(x^{\text{lth } x}).
\]

$x \in B$ means: $x^{\text{lth } x} \in A$ and, for every $n < \text{lth } x$, $((x)_n)$, is the ‘witnessing information’ that $x^n \in A$.

One easily derives Tree($B$), Bar($B, Q$), Mon($B, Q$), Ind($B, Q$); hence by 3.2.13 (observe that $B \in \mathcal{L}^-(\text{TAPP})$) $Q(\ )$, so $P(\ )$ (for $\langle \rangle^0 = (\ )$). □

3.3. Inductively defined functionals

Here we establish the properties of $\text{L}_0, I_i$ that are needed in 3.2 and 3.4. For this, we define the theories $T^*_2$ and $T_2$.

3.3.1. Definition. $T^*_2$ is obtained from $T^*_1$ by omitting the axioms ECS1–4; if we also drop the sequences variables $\alpha, \beta, \ldots$, their axioms and rules, and replace $\alpha, \beta$ in $\text{L}_0\text{AX}1, 3$ and $I_i\text{AX}$ by the object variables $a, b$, we get the theory $T_2$. So $T_2$ is an extension of $\text{TAPP} + \text{EAC}$ with tree variables and inductively defined sets of functionals.
3.3.2. Lemma. $T^*_2 \vdash A \Rightarrow T_2 \vdash A^-$, where $\rightarrow: T^*_2 \rightarrow T_2$ is the extension of the mapping $\rightarrow$ of 3.1.3 to $T^*_2$.

Proof. As in 3.1.3. □

3.3.3. Corollary. $T^*_2$ is conservative over $T_2$.

3.3.4. Lemma. In $T^*_2$ we have:

(i) $\alpha \equiv \beta \land \alpha \in \bar{T} \land \phi \in I_0(T) \rightarrow \phi \alpha = \phi \beta$.

(ii) $\alpha \equiv \beta \land \alpha \in \bar{T} \land f \in I_1(T, S) \rightarrow f \alpha = f \beta$.

(iii) $\phi =_T \psi \land \phi \in I_0(T) \rightarrow \psi \in I_0(T)$.

(iv) $f =_T g \land f \in I_1(T, S) \rightarrow g \in I_1(T, S)$.

(v) $S \subseteq T \rightarrow I_0(T) \subseteq I_0(S)$.

(vi) $S_1 \subseteq T_1 \land T_2 \subseteq S_2 \rightarrow I_1(T_1, T_2) \subseteq I_1(S_1, S_2)$.

(vii) $\phi \in I_0(T) \rightarrow \phi =_T \phi_\langle \rangle$.

Proof. (i), (iii) and (v) are proved using I₀AX3, taking for $\phi \in P(x)$ respectively

$$\forall \alpha \beta \ (\alpha \in \bar{T} \land \alpha \equiv \beta \rightarrow \phi \alpha = \phi \beta),$$

$$\forall \psi \ (\psi =_T \phi \rightarrow \psi \in I_0(T_x)) \text{ and } \phi \in I_0(S_x);$$

also TRAX8 is used.

(ii), (iv), (vi) follow from (i), (iii), (v) and the definition of $I_1$.

(vii) follows from (i) (for $\alpha = \langle \rangle * \alpha$). □

3.3.5. Lemma. (i) $T^*_2 \vdash \phi \in I_0(T) \iff \forall y \in T (\text{ith } y = n \rightarrow \phi_y \in I_0(T_y))$.

(ii) Let $\phi \in I_0(T)$, $\phi \in T \Rightarrow \mathbb{N}$. Then

$$\forall \psi (\forall \alpha \in \bar{T} (\psi_{\alpha(\phi_a)} \in I_0(T_{\alpha(\phi_a)})) \iff \psi \in I_0(T)).$$

Proof. (i) The case $n = 0$ follows from $\text{ith } y = 0 \iff y \sim \langle \rangle$, $T = T_{\langle \rangle}$ and $\phi =_T \phi_\langle \rangle$.

For $n = 1$, $\leftarrow$ follows with I₀AX2, $\rightarrow$ with I₀AX3 where $\phi \in P(x) := \forall y \in T_x (\phi_y \in I_0(T_{x * y}))$. For $n > 1$, use induction over $\mathbb{N}$.

(ii) Use (i) and I₀AX3 with

$$\phi \in P(x) := \phi \in (\bar{T} \Rightarrow \mathbb{N}) \rightarrow \forall \psi (\forall \alpha \in \bar{T} (\psi_{\alpha(\phi_a)} \in I_0(T_{x * \alpha(\phi_a)})) \rightarrow \psi \in I_0(T_x)).$$

3.3.6. Definition. We define

$$\alpha \times \beta := \lambda n \cdot (\alpha n, \beta n)$$

$$\pi_i := \lambda a n \cdot (an)_i \quad (i = 0, 1)$$

$$x^* := \lambda \alpha \cdot x \star \alpha$$

$$f \otimes g := \lambda a n \cdot (fan, gan)$$

$$f \circ g := \lambda \alpha \cdot f(g\alpha).$$
3.3.7. Lemma. (i) $\forall n (\lambda \alpha \cdot \alpha n \in I_0(T))$.
(ii) $\forall \phi \in I_0(T) \forall x (\lambda \alpha \cdot x(\phi \alpha) \in I_0(T))$.
(iii) $\forall \phi, \psi \in I_0(T)(\lambda \alpha \cdot (\phi \alpha, \psi \alpha) \in I_0(T))$.

Proof. (i) Induction over $\mathbb{N}$, using I$_{0}$AX1, 2.
(ii) Induction over $I_0$.
(iii) Double induction over $I_0$. □

3.3.8. Lemma. (i) $\lambda \alpha \cdot \phi \alpha + 1 \in I_0(T)$.
(ii) $\lambda \alpha \cdot \alpha \in I_1(T)$.
(iii) $\phi, \psi \in I_0(T) \rightarrow \lambda \alpha \cdot \max(\phi \alpha, \psi \alpha) \in I_0(T)$.
(iv) $x \in T \rightarrow x* \in I_1(T, T)$.
(v) $\pi_i \in I_1(T_0 \times T_1, T_i) (i = 0, 1)$
(vi) $\forall f \in I_1(S, T_1) \forall g \in I_1(S, T_2)(f \otimes g \in I_1(S, T_1 \times T_2))$.

Proof. (i) By 3.3.7(ii).
(ii) By 3.3.7(i) and the definition of $I_1$.
(iii) Combine 3.3.7(i), (ii).
(iv) Use I$_{0}$AX1, 3.3.7(i) and the definition of $I_1$.
(v) By 3.3.7(i) and the definition of $I_1$.
(iv) By 3.3.7(iii). □

For the important Lemma 3.3.11 we need not only to know that all $\phi \in I_0(T)$
are continuous, but also that any such $\phi$ has a modulus $\delta \in I_0(T) \cap (\bar{T} \Rightarrow \mathbb{N})$ which
is also its own modulus; analogously for $I_1(S, T)$.

3.3.9. Definition. Let $\phi \in I_0(T), f \in I_1(S, T)$.
(i) $\delta \bmod \phi := \delta \in (\bar{T} \Rightarrow \mathbb{N}) \land \forall \alpha \beta \in \bar{T} (\alpha(\delta \alpha) = \bar{\beta}(\delta \alpha) \rightarrow \phi \alpha = \phi \beta)$.
(ii) $\delta \in M_0(T) := \delta \in I_0(T) \land \delta \bmod \delta$.
(iii) $d \bmod f := d \in (\mathbb{N} \Rightarrow (\bar{S} \Rightarrow \mathbb{N})) \land \forall n \forall \alpha \beta \in \bar{S} (\bar{\alpha}(dn \alpha) = \bar{\beta}(dn \alpha) \rightarrow d \alpha n = d \beta n)$.
(iv) $d \in M_1(S) := \forall n (dn \in M_0(S))$.

3.3.10. Lemma. (i) $\forall \phi \in I_0(T) \exists \delta \in M_0(T)(\delta \bmod \phi)$.
(ii) $\forall f \in I_1(S, T) \exists d \in M_1(S)(d \bmod f)$.

Proof. (i) Use I$_{0}$AX3 with $\phi \in P(x) := \exists \delta \in M_0(T_0)(\delta \bmod \phi)$.
- $\forall \alpha \in T_0 (\phi \alpha = y)$: take $\delta := \lambda \alpha \cdot 0$.
- Assume $\forall \hat{y} \in T_0 \exists \delta \in M_0(T_0, y)(\delta \bmod \phi)$.
- By EAC:

$\exists D \forall \hat{y} \in T_0 (Dy \in M_0(T_0, y) \land Dy \bmod \phi)$.

Define

$\delta' := \lambda \alpha \cdot D(\alpha 0)(\lambda n \cdot \alpha(n + 1)) + 1$,

then $\delta' \in I_0(T_0)$ (by I$_{0}$AX2 and 3.3.8(i)), $\delta'$ $\bmod \phi$ and $\delta'$ $\bmod \delta'$. 

(ii) Assume \( f \in I_1(S, T) \), so by the definition of \( I_1 \) we have \( \forall n \) \((\lambda \alpha \cdot f \alpha n \in I_0(S))\). With (i):

\[ \forall n \exists \delta \in M_0(S)(\delta \text{ mod } \lambda \alpha \cdot f \alpha n) \]

using EAC, we find some \( D \) with

\[ \forall n \ (Dn \in M_0(S) \land Dn \text{ mod } \lambda \alpha \cdot f \alpha n). \]

Now define \( d \) by

\[
\begin{align*}
\{ &d0 := D0, \\
&(d(n + 1) := \lambda \alpha \cdot \max(\,dn, D(n + 1) \alpha),
\}
\]

then \( d \in M_1(S) \) (by 3.3.8(iii), induction over \( n \)) and \( d \text{ Mod } f \). \( \Box \)

3.3.11. Lemma (Closure of \( I_0 \)- and \( I_1 \)-sets under composition).

(i) \( \forall \phi \in I_0(S) \forall f \in I_1(S, T)(\phi \circ f \in I_0(T)) \).

(ii) \( \forall f \in I_1(S, T) \forall g \in I_1(S', S)(f \circ g \in I_1(S', T)) \).

Proof. (i) We use \( I_0AX3 \) with \( \psi \in P(\alpha) := \forall T \forall f \in I_1(T, S)\phi \; (\phi \cdot f \in I_0(T)) \).

- \( \forall \alpha \in S_\lambda (\phi \alpha = y) \): then \( \phi \cdot f \) is also constant on \( T \), and (by \( I_0AX1 \)) in \( I_0(T) \).

- Assume

\[ (1) \quad \forall y \in S_\lambda \forall T \forall f \in I_1(T, S_{\alpha \cdot \lambda})(\psi \cdot f \in I_0(T)) \]

and let \( g \in I_1(T, S) \). Then \( \lambda \alpha \cdot g \alpha 0 \in I_0(T), \) so by 3.3.10(ii), \( \delta \text{ mod } \lambda \alpha \cdot g \alpha 0 \) for some \( \delta \in M_0(T) \). Now let \( \alpha \in \hat{T} \) be arbitrary and define \( z := g \alpha 0. \) Then

\[ \forall \beta \in T_{\bar{\alpha}(\delta \alpha)}(g(\bar{\alpha}(\delta \alpha) \ast \beta)0 = z). \]

Define

\[ h := \lambda \beta n \cdot g(\bar{\alpha}(\delta \alpha) \ast \beta)(n + 1), \]

then, by 3.3.7(i), for all \( n \)

\[ \lambda \beta \cdot h \beta n = \lambda \beta \cdot g(\bar{\alpha}(\delta \alpha) \ast \beta)(n + 1) \in I_0(T_{\bar{\alpha}(\delta \alpha)}), \]

so \( h \in I_1(T_{\bar{\alpha}(\delta \alpha)}, S_\lambda) \) by the definition of \( I_1 \). Now

\[ \phi \circ h = \lambda \beta \cdot \phi((g(\bar{\alpha}(\delta \alpha) \ast \beta)0 \ast \lambda n \cdot g(\bar{\alpha}(\delta \alpha) \ast \beta)(n + 1)) \]

\[ = \lambda \beta \cdot \phi(g(\bar{\alpha}(\delta \alpha) \ast \beta)) = (\phi \circ g)(\bar{\alpha}(\delta \alpha)). \]

By (1), \( \phi \circ h \in I_0(T_{\bar{\alpha}(\delta \alpha)}), \) so with 3.3.5(ii) we have \( \phi \circ g \in I_0(T) \).

(ii) Easy, use \( \lambda \alpha \cdot (f \circ g) \alpha n = (\lambda \alpha \cdot (f \alpha) \alpha n) \circ g, \) (i) and the definition of \( I_1 \). \( \Box \)

3.3.12. Lemma. Let \( \delta \in M_0(T) \), and let \( A \) satisfy

(ii) \( \forall x \in T \forall pq (\forall \alpha \in \bar{\alpha}_{\delta}(p \alpha = q \alpha) \rightarrow (A(x, p) \leftrightarrow A(x, q))). \)
Then:
(i) $\forall \alpha \in \bar{T} \exists \phi \in I_0(T) A(\bar{\alpha}(\delta \alpha), \phi) \rightarrow \exists \psi \in I_0(T) \forall \alpha \in \bar{T} A(\bar{\alpha}(\delta \alpha), \psi_{\bar{\alpha}(\delta \alpha)})$.
(ii) $\forall \alpha \in \bar{T} \exists f \in I_1(T, S) A(\bar{\alpha}(\delta \alpha), f) \rightarrow \exists g \in I_1(T, S) \forall \alpha \in \bar{T} A(\bar{\alpha}(\delta \alpha), g_{\bar{\alpha}(\delta \alpha)})$.

Proof. (i) Assume $\forall \alpha \in \bar{T} \exists \phi \in I_0(T) A(\bar{\alpha}(\delta \alpha), \phi)$. Using EAC, we find a $\Phi$ with

$$(3) \quad \forall \alpha \in \bar{T} (\Phi \alpha \in I_0(T_{\bar{\alpha}(\delta \alpha)}) \wedge A(\bar{\alpha}(\delta \alpha), \Phi \alpha)).$$

We also have, by 3.2.5, $\forall x \in T \exists \beta (\beta \in \bar{T})$; EAC gives us an $F$ with $\forall x \in T (Fx \in \bar{T})$, i.e., $\forall x \in T (x \ast Fx \in \bar{T})$.

Now define, for $\alpha \in \bar{T}$:

$$\alpha_{\delta} := \bar{\alpha}(\delta \alpha) \ast F(\bar{\alpha}(\delta \alpha)),$$

then $\alpha_{\delta} \in \bar{T}$ and $\delta(\alpha_{\delta}) = \delta \alpha$ (for $\delta \mod \delta$) so $\bar{\alpha}_{\delta}(\delta(\alpha_{\delta})) = \bar{\alpha}(\delta \alpha)$; also by $\delta \mod \delta$

$$(4) \quad \forall \beta \in T_{\bar{\alpha}(\delta \alpha)}((\bar{\alpha}(\delta \alpha) \ast \beta)_{\delta} = \alpha_{\delta}).$$

Define

$$\psi := \lambda \alpha, (\Phi \alpha_{\delta})(\lambda n, \alpha(n + \delta \alpha)),$$

then, by (4),

$$(5) \quad \forall \beta \in T_{\bar{\alpha}(\delta \alpha)}(\psi_{\bar{\alpha}(\delta \alpha)} \beta = \Phi \alpha_{\delta} \beta).$$

Now (3) gives

$$\forall \alpha \in \bar{T} (\Phi \alpha_{\delta} \in I_0(T_{\bar{\alpha}(\delta \alpha)}) \wedge A(\bar{\alpha}(\delta \alpha), \Phi \alpha_{\delta}))$$

so, with (2) and (5)

$$\forall \alpha \in \bar{T} (\psi_{\bar{\alpha}(\delta \alpha)} \in I_0(T_{\bar{\alpha}(\delta \alpha)}) \wedge A(\bar{\alpha}(\delta \alpha), \psi_{\bar{\alpha}(\delta \alpha)})).$$

With 3.3.5(ii):

$$\exists \psi \in I_0(T) \forall \alpha \in \bar{T} A(\bar{\alpha}(\delta \alpha), \psi_{\bar{\alpha}(\delta \alpha)}).$$

(ii) Analogously. \qed

3.3.13. Lemma. Let $\varepsilon \in M_0(T)$, $f \in I_1(S, T)$, $\alpha \in \bar{S}$. Then

$$\exists \delta \in M_0(S) \forall \beta \in \bar{S}_{\bar{\alpha}(\delta \alpha)}(f(\bar{\alpha}(\delta \alpha) \ast \beta) \in \bar{f}(\varepsilon(f \alpha))).$$

Remark. The existence of $\delta$ follows from the continuity of $\varepsilon$ and $f$; $\delta \in M_0(S)$ requires a more subtle argument.
Proof. \( f \in I_1(S, T) \) implies (by 3.3.10(ii)) \( d \text{ Mod } f \) for some \( d \in M_1(S) \), so
\[
\forall n \ \forall \alpha \in \hat{S} \ \forall \beta \in \hat{S}_{\delta(\alpha n)}(f(\bar{\alpha}(\alpha n) \cdot \beta) \in \overline{f(\alpha)}).
\]
Define
\[
\delta := \lambda \alpha \cdot d(\varepsilon(f\alpha))\alpha,
\]
then
\[
\forall \alpha \in \hat{S} \ \forall \beta \in \hat{S}_{\delta(\alpha)}(f(\bar{\alpha}(\delta \alpha) \cdot \beta) \in \overline{f(\varepsilon(f\alpha))}).
\]
It remains to be shown that \( \delta \in I_0(S) \) and \( \delta \text{ mod } \delta \). Now \( \varepsilon \in M_0(T) \), \( f \in I_1(S, T) \), so \( \varepsilon \circ f \in I_0(S) \); let \( \eta \in M_0(S) \), \( \eta \text{ mod } \varepsilon \circ f \) (using 3.3.10(i)), then
\[
(6) \ \forall \alpha \in \hat{S} \ \exists n \ \forall \beta \in \hat{S}_{\delta(\eta \alpha)}(\varepsilon \circ f)(\bar{\alpha}(\eta \alpha) \cdot \beta) = n.
\]
Now, by definition of \( \delta \)
\[
\forall \alpha \in \hat{S} \ [\delta_{\alpha(\eta \alpha)} = \lambda \beta \cdot d((\varepsilon \circ f)(\bar{\alpha}(\eta \alpha) \cdot \beta))(\bar{\alpha}(\eta \alpha) \cdot \beta)]
\]
so, by (6)
\[
\forall \alpha \in \hat{S} \ \exists n \ [\delta_{\alpha(\eta \alpha)} = \lambda \beta \cdot d(\bar{\alpha}(\eta \alpha) \cdot \beta) = (dn)_{\delta(\eta \alpha)}].
\]
By 3.3.5(i) we get \( \forall \alpha \in \hat{S} \ (\delta_{\alpha(\eta \alpha)} \in I_0(S_{\delta(\eta \alpha)}) \) and with 3.3.5(ii) this gives \( \delta \in I_0(S) \).
To see that \( \delta \text{ mod } \delta \), assume \( \bar{\alpha}(\delta \alpha) = \bar{\beta}(\delta \alpha) \), i.e.
\[
(7) \ \bar{\alpha}(d(\varepsilon(f\alpha))\alpha) = \bar{\beta}(d(\varepsilon(f\alpha))\alpha).
\]
\( d \in M_1(S) \), so \( \forall n \ (dn \in M_0(S)) \), hence (7) implies
\[
(8) \ d(\varepsilon(f\alpha))\alpha = d(\varepsilon(f\beta))\beta.
\]
Also \( d \text{ Mod } f \), so with (8)
\[
\overline{f(\alpha)}(\varepsilon(f\alpha)) = \overline{f(\beta)}(\varepsilon(f\beta));
\]
with \( \varepsilon \text{ mod } \varepsilon \) this yields \( \varepsilon(f\alpha) = \varepsilon(f\beta) \). Combining this with (8), we conclude
\[
d(\varepsilon(f\alpha))\alpha = d(\varepsilon(f\beta))\beta, \text{ i.e., } \delta \alpha = \delta \beta. \quad \Box
\]

3.4. Forcing (alias elimination)

3.4.1. In this section we interpret \( T^*_1 \) in \( T_2 \). This interpretation is presented in two ways: first as an elimination translation (in the sense of [20] and [28]), which is somewhat easier to understand, then as a definition of forcing, which has a more semantic flavour.

3.4.2. To describe the elimination translation, we consider \( \forall \alpha \in \hat{S}, \exists \beta \in \hat{T} \) as quantifiers, not as abbreviations of \( \forall \alpha \ (\alpha \in \hat{S} \rightarrow \cdots) \) etc; \( \forall \alpha, \exists \beta \) are read as
∀α ∈ Ü, ∃β ∈ Ü. Also ∀m, ∃n are considered as quantifiers ranging over N. Now the elimination translation for formulae without free sequence variables reads

\[ P \vdash P \quad (P \text{ prime}) \]
\[ (A \land B) \vdash (A \land B), \quad (A \rightarrow B) \vdash (A \rightarrow B) \]
\[ (\forall x A) \vdash (\forall x A), \quad (\exists x A) \vdash (\exists x A) \]
\[ (\forall n A) \vdash (\forall n A), \quad (\exists n A) \vdash (\exists n A) \]
\[ \forall T A \vdash \forall T A, \quad \exists T A \vdash \exists T A \]
\[ \forall \alpha \in \hat{T} (A \land B) \vdash (\forall \alpha \in \hat{T} A) \land (\forall \alpha \in \hat{T} B) \]
\[ \forall \alpha \in \hat{T} (A \rightarrow B) \vdash \forall S \forall \phi \in I_1(S, T) \]
\[ \vdash (\forall \alpha \in \hat{T} f(\alpha) \rightarrow \forall \alpha \in \hat{T} f(\alpha)) \]
\[ \forall \alpha \in \hat{T} \forall x A \vdash \forall x \forall \alpha \in \hat{T} A \]
\[ \forall \alpha \in \hat{T} \exists x A \vdash \exists \phi \in I_0(T) (\forall \alpha \in \hat{T} A(\phi \alpha)) \]
\[ \forall \alpha \in \hat{T} \forall n A \vdash \forall n \forall \alpha \in \hat{T} A \]
\[ \forall \alpha \in \hat{T} \exists n A \vdash \exists \phi \in I_0(T) \cap (\hat{T} \rightarrow N) (\forall \alpha \in \hat{T} A(\phi \alpha)) \]
\[ \forall \alpha \in \hat{T} \forall \beta \in \hat{S} A(\alpha, \beta) \vdash \forall f \in I_1(T \times S, T) \forall g \in I_1(T \times S, S) \]
\[ \forall \alpha \in \hat{T} \forall \beta \in \hat{S} A(\alpha, \beta) \rightarrow \forall f \in I_1(T \times S, T) \forall \alpha \in \hat{T} A(\alpha, g \alpha) \]
\[ \forall \alpha \in \hat{T} \exists \beta \in \hat{S} A(\alpha, \beta) \rightarrow \exists g \in I_1(T, S) (\forall \alpha \in \hat{T} A(\alpha, g \alpha)) \]
\[ \forall \alpha \in \hat{T} \forall S A \vdash \forall S (\forall \alpha \in \hat{T} A) \]
\[ \forall \alpha \in \hat{T} \exists S A \vdash \exists S (\forall \alpha \in \hat{T} A) \]
\[ \forall \alpha \in \hat{T} A \vdash \exists a \in \hat{T} A \]

A few examples:

(i) \[ (\forall \alpha \land \forall n \exists x (an = x)) \]

\[ \vdash \forall n \land \forall \alpha \exists x (an = x) \]
\[ = \forall n \land \forall \alpha \exists x (an = x) \]
\[ = \forall n \exists \phi \in I_0(U) (\forall \alpha (an = \phi \alpha)) \]
\[ = \forall n \exists \phi \in I_0(U) \forall a (an = \phi a); \]

using 3.3.7(i) and 3.3.2, we see that this interpretation of SEQAX1 is true in T₂.

(ii) \[ (\forall x \exists \alpha \land \forall n (xn = an)) \]

\[ = \forall x \exists \alpha \land \forall n (xn = an), \]

which is also true in T₂.

(iii) \[ (\forall \alpha \exists x \land \forall n (an = xn)) \]

\[ = \exists \phi \in I_0(U) (\forall \alpha \land \forall n (an = \phi an)) \]
\[ = \exists \phi \in I_0(U) \forall n \forall \alpha (an = \phi an), \]
and this is definitely not true in $T_2$, for by 3.3.2 and 3.3.10(i) the value of $\phi a$ is completely determined by an initial segment of $a$.

3.4.3. Now we turn to forcing. First we introduce the concept of distinguished terms of some formula $A$: these are certain term occurrences in $A$, usually indicated by $\bar{p} = (p_1, \ldots, p_n)$. Sometimes they are underlined to distinguish them, and we write $A = A(\bar{p})$ or $A = A(\bar{p})$. This concept is needed for the following important definition.

3.4.4. Definition. Let $A$ be a formula with distinguished terms $\bar{p}$, and let $f$ be some term. The restriction of $A$ along $f$ is defined by

$$A \upharpoonright f := A[\bar{p} := \bar{p} \upharpoonright f],$$

where $\bar{p} \upharpoonright f$ stands for $p_1 \circ f, \ldots, p_n \circ f$; they are exactly the distinguished terms of $A \upharpoonright f$.

3.4.5. Examples. (i) $A \upharpoonright f = A$ if $A$ contains no distinguished terms.

(ii) $(\phi a = \psi b) \upharpoonright f = (\phi \circ f)a = (\psi \circ f)b$.

3.4.6. In the definition of forcing we shall give in a moment, we associate to every formula $A$ of $T_2^*$ and tree variable $T$ a formula $T \models A$ ($T$ forces $A$) of $T_2$. If $A$ contains the choice variables $\alpha_1, \ldots, \alpha_n$, free, then we associate the free TAPP-variables $f_1, \ldots, f_n$ to $\alpha_1, \ldots, \alpha_n$ and put

$$T \models A(\alpha_1, \ldots, \alpha_n) := \forall S(f_1, \ldots, f_n \in I_1(S, T) \rightarrow S \models A(f_1, \ldots, f_n)).$$

For formulae without free sequence variables and with distinguished terms $\bar{p}$, we define

$$T \models P \quad := \forall a \in \bar{T} (P[\bar{p} := \bar{p}a]) \quad \text{for prime } P$$

$$T \models A \land B \quad := (T \models A) \land (T \models B)$$

$$T \models A \rightarrow B \quad := \forall S \forall f \in I_1(S, T)(S \models (A \upharpoonright f) \rightarrow S \models (B \upharpoonright f))$$

$$T \models \forall x A \quad := \forall x (T \models A)$$

$$T \models \exists x A(x) \quad := \exists \phi \in L_0(T)(T \models A(\phi))$$

$$T \models \forall n A \quad := \forall n (T \models A)$$

$$T \models \exists n A \quad := \exists \phi \in L_0(T) \cap (\bar{T} \Rightarrow \forall n)(T \models A(\phi))$$

$$T \models \forall S A \quad := \forall S (T \models A)$$

$$T \models \exists S A \quad := \exists S (T \models A)$$

$$T \models \forall \alpha A(\alpha) := \forall S \forall f \in I_1(S, T) \forall g \in I_1(S, U)(S \models (A \upharpoonright f)(g))$$

$$N.B. \quad (A \upharpoonright f)(g) \text{ is to be read as } (A \upharpoonright f)[\alpha := g]$$

$$T \models \exists \alpha A(\alpha) := \exists g \in I_1(T, U)(T \models A(g)).$$
3.4.7. Examples.

(i) \[ T \vdash \text{SEQAX1} = T \vdash \forall \alpha \forall n \exists x \ (an = x)\]
\[= \forall S \forall f \in I_1(S, T) \forall g \in I_1(S, U)(S \vdash (\forall n \exists x (an = x))[\alpha := g])\]
\[= \forall S \forall g \in I_1(S, U)(S \vdash (\forall n \exists x (an = x))[\alpha := g])\]
\[= \forall S \forall g \in I_1(S, U)(S \vdash \forall n \exists x (gn = x))\]
\[= \forall S \forall g \in I_1(S, U) \forall n \exists \phi \in I_0(S) S \vdash (\forall n \exists x (an = x))[\alpha := g]\]
\[= \forall S \forall g \in I_1(S, U) \forall n \exists \phi \in I_0(S) \forall a \in S \ (gan = \phi a).\]

(ii) \[ T \vdash \text{SEQAX2} = T \vdash \forall x \exists \alpha \forall n \ (xn = an)\]
\[= \forall X T \vdash \exists \alpha \forall n \ (xn = an)\]
\[= \forall X \exists g \in I_1(T, U) T \vdash \forall n \ (xn = gn)\]
\[= \forall X \exists g \in I_1(T, U) \forall n \forall a \in \tilde{T} \ (xn = gan).\]

(iii) \[ T \vdash \forall \alpha \exists x \forall n \ (an = xn)\]
\[= \forall S \forall f \in I_1(S, T) \forall g \in I_1(S, U) S \vdash (\exists x \forall n (an = xn)) [\alpha := g]\]
\[= \forall S \forall g \in I_1(S, U) S \vdash \exists x \forall n (gn = xn)\]
\[= \forall S \forall g \in I_1(S, U) \exists \phi \in I_0(S) S \vdash \forall n \ (gn = \phi n)\]
\[= \forall S \forall g \in I_1(S, U) \exists \phi \in I_0(S) \forall a \in \tilde{S} \ (gan = \phi an).\]

To show that forcing and the elimination translation are equivalent interpretations, we need the so-called monotonicity property of \(\vdash\) (proved in 3.4.10), and 3.4.12(iii).

3.4.8. Lemma. For totally regular formulae \(A\) we have
\[\widetilde{T}_2 \vdash T \vdash A(\tilde{\alpha}) \leftrightarrow \forall \alpha \in \tilde{T} A(\tilde{\alpha})\]

Proof. Formula induction. Most cases are trivial or easy, except \(A = \forall \beta \in \overline{T'} B(\tilde{\alpha}, \beta)\). By 3.4.12(iii), \(T \vdash \forall \beta \in \overline{T'} B(\tilde{\alpha}, \beta)\) is equivalent to

(1) \[ \forall S \forall f \in I_1(S, T) \forall g \in I_1(S, T') (S \vdash B(\tilde{\alpha} f, g)); \]
also
\[ \forall \alpha \in \tilde{T} \forall \beta \in \overline{T'} B(\tilde{\alpha} \beta, \beta) \]
\[= \forall f \in I_1(T \times T', T) \forall g \in I_1(T \times T', T') \forall \alpha \in \tilde{T} \times \tilde{T'} B(\tilde{\alpha} f, g \tilde{\alpha} \beta),\]
which is equivalent to

(2) \[ \forall f' \in I_1(T \times T', T) \forall g' \in I_1(T \times T', T') (T \times T' \vdash B(g' \tilde{\alpha} f', g \tilde{\alpha} \beta)). \]

(1) \(\rightarrow\) (2) is evident: take \(S := T \times T'\). For (2) \(\rightarrow\) (1) we argue as follows. By 3.4.10, (2) implies

(3) \[ \forall f' \in I_1(T \times T', T) \forall g' \in I_1(T \times T', T') \forall \forall h \in I_1(S, T \times T') \]
\[ (S \vdash B(g' \tilde{\alpha} h, \tilde{\alpha} f' \tilde{\alpha} h)).\]
Now take $h := f \otimes g$, $f' := \pi_0$, $g' := \pi_1$. Use $\pi_0 \circ (f \otimes g) \equiv_{S} f$, $\pi_1 \circ (f \otimes g) \equiv_{S} g$, and we get (1). □

We shall now prove some lemmata needed for the soundness theorem for $\Gamma$.  

3.4.9. **Lemma** (Substitution). (i) \( p =_{T} q \rightarrow (T \models A(p) \iff T \models A(q)) \).

(ii) \( T \models A(\tau) \iff T \models A(\lambda \alpha \cdot \tau), \ \tau \text{ a term of } \mathcal{L}(\text{TAPP}) \).

**Proof.** Straightforward, with formula induction. □

3.4.10. **Lemma** (Monotonicity). \( T \models A \leftrightarrow \forall S \forall f \in I_1(S, T)(S \models (A \uparrow f)) \).

**Proof.** \( \rightarrow \) follows from \( \lambda \alpha \cdot \alpha \in I_1(T, T) \) (3.3.8(ii)).

\( \rightarrow \) is proved with formula induction: as an example, we treat the cases \( A = \exists x B \) and \( A = \forall \alpha B \).

\( A = \exists x B(x) \): assume \( T \models \exists x B(x) \), i.e.

\[ \exists \phi \in I_0(T) \ T \models B(\phi). \]

By induction hypothesis:

\[ \exists \psi \in I_0(T) \forall S \forall f \in I_1(S, T) \ (S \models (B \uparrow f)(\psi \uparrow f)) \]

so, with Lemma 3.3.11(i):

\[ \forall S \forall f \in I_1(S, T) \exists \psi \in I_0(S) \ (S \models (B \uparrow f)(\psi)) \]

i.e., \( \forall S \forall f \in I_1(S, T) \exists \psi \in I_0(S) \ (S \models (B \uparrow f)(\psi)) \). \( A = \forall \alpha B(\alpha) \): assume \( T \models \forall \alpha B(\alpha) \), i.e.

\[ \forall S \forall f \in I_1(S, T) \forall g \in I_1(S, U) \ (S \models (B \uparrow f)(g)) \]

with Lemma 3.3.11(ii):

\[ \forall S' \forall f' \in I_1(S', T) \forall S \forall f \in I_1(S, S') \forall g \in I_1(S, U) \ (S \models (B \uparrow f')(g)) \]

i.e., \( \forall S' \forall f' \in I_1(S', T) \ (S' \models (\forall \alpha B(\alpha) \uparrow f')) \). □

3.4.11. **Lemma** (Bar-property). \( \forall \delta \in M_0(T) \forall a \in \tilde{T} \ (T_{\tilde{a}(\delta a)} \models (A \uparrow \tilde{a}(\delta a) *)) \iff T \models A \).

**Proof.** \( \rightarrow \) follows from the previous lemma and Lemma 3.3.8(iv).

\( \rightarrow \) requires formula induction: we consider the key cases \( A = B \rightarrow C \), \( A = \exists x B \), \( A = B \rightarrow C \): assume \( \delta \in M_0(T) \) and \( \forall a \in \tilde{T} \ (T_{\tilde{a}(\delta a)} \models ((B \rightarrow C) \uparrow \tilde{a}(\delta a) *)) \), i.e.

1. \( \forall a \in \tilde{T} \forall S \forall f \in I_1(S, T_{\tilde{a}(\delta a)})(S \models (B \uparrow (\tilde{a}(\delta a) *) \circ f) \rightarrow S \models (C \uparrow (\tilde{a}(\delta a) *) \circ f)) \)

and let \( g \in I_1(S, T) \), \( b \in \tilde{S} \). By Lemma 3.3.13:

2. \( \exists \eta \in M_0(S) \forall a \in \tilde{S}_{\tilde{a}(\eta a)}(g(\tilde{b}(\eta b) \ast a) \in \overline{gb}(\delta(gb))). \)
Define \( h \) by
\[
h := \lambda n . g(b(\eta b) * a)(n + \delta(g b)),
\]
then \( \overline{g b(b(\delta g b))} \circ h = g \circ (b(\eta b) *) \) (by (2)) and \( h \in I_1(S_{b(\eta b)}) \), \( T_{g b(b(\delta g b))} \) (by 3.3.7(i), 3.3.8(iv), 3.3.11(ii)). So, by (1) \( (a := g b, \ S := S_{b(\eta b)}, \ f := h) \):

\[
(3) \quad S_{b(\eta b)} \models (B \uparrow g \circ (b(\eta b) *)) \rightarrow S_{b(\eta b)} \models (C \uparrow g \circ (b(\eta b) *)).
\]

Since we also have (Lemma 3.4.10 with \( f := (b(\eta b) *) \))

\[
(4) \quad S \models B \uparrow g \rightarrow S_{b(\eta b)} \models (B \uparrow g \circ (b(\eta b) *))
\]

and, by induction hypothesis,

\[
(5) \quad \forall b \in S (S_{b(\eta b)} \models (C \uparrow g \circ (b(\eta b) *)) \rightarrow S \models C \uparrow g).
\]

we get (combining (3), (4), (5))

\[
\forall S \forall g \in I_1(S, T)(S \models B \uparrow g \rightarrow S \models C \uparrow g)
\]
i.e., \( S \models B \rightarrow C \).

\[
A = \exists x \ b(x): \text{ assume } \delta \in M_0(T) \text{ and } \forall a \in \tilde{T} (T_{\tilde{a}(\delta a)} \models (\exists x \ b(x) \uparrow (a(\delta a) *))),
\]
i.e.

\[
\forall a \in \tilde{T} \exists \phi \in I_0(T_{\tilde{a}(\delta a)})(T_{\tilde{a}(\delta a)} \models (B \uparrow (a(\delta a) *))((\phi))).
\]

By 3.3.12(i) and 3.4.9(i):

\[
\exists \psi \in I_0(T) \forall a \in \tilde{T} (T_{\tilde{a}(\delta a)} \models (B \uparrow (a(\delta a) *))((\psi \circ (a(\delta a) *)))).
\]

With the induction hypothesis:

\[
\exists \psi \in I_0(T)(T \models B(\phi)),
\]
i.e., \( T \models \exists x \ b(x) \).

3.4.12. Lemma. (i) \( T \models \forall n \ A n \leftrightarrow T \models \exists x \ (x \in N \rightarrow A x) \).
(ii) \( T \models \exists n \ A n \leftrightarrow T \models \exists x \ (x \in N \land Ax) \).
(iii) \( T \models \forall a \in S \ A a \leftrightarrow \forall T' \forall f \in I_1(T', T) \forall g \in I_1(T', T)(T' \models (A \uparrow f)(g)) \).
(iv) \( T \models \exists a \in S \ A a \leftrightarrow \exists g \in I_1(T, S)(T \models A(g)) \).
(v) \( T \models A(f g) \leftrightarrow T \models A(f \circ g) \).
(vi) \( T \models A \rightarrow T \models B \uparrow f \rightarrow T \models B \uparrow f \).
(vii) \( T \models A \rightarrow T \models A \).
(viii) If \( A \) contains no free sequence variables and no distinguished terms, then:
(a) \( S \models A \leftrightarrow T \models A \).
(b) \( T \models \exists x \ A \leftrightarrow \exists x \ (T \models A) \).
(viii) If \( A \in \mathcal{L}(T_2) \), then \( T \models A \leftrightarrow A \).

Proof. (i), (ii) Easy, write out the definition of \( T \models \forall x \cdots \), \( T \models \exists x \cdots \) and use 3.4.10.
(iii) $\forall \alpha \in \mathcal{A} \alpha$ abbreviates $\forall \alpha (\forall n (an \in S) \rightarrow A \alpha)$, so writing out $T \Vdash \forall \alpha \in \mathcal{A} A \alpha$ yields

$$\forall T' \forall f \in I_1(T', T) \forall g \in I_1(T', U) \forall T'' \forall h \in I_1(T'', T')$$

$$(\forall n \forall \alpha \in \mathcal{T}'' (g(\alpha)n \in S) \rightarrow T'' \Vdash ((A \mapsto f)(g) \mapsto h));$$

this is equivalent to (use 3.3.4(vi), 3.3.11(ii))

$$\forall T' \forall f \in I_1(T', T) \forall g \in I_1(T', U) \forall T'' \forall h \in I_1(T'', T')$$

$$(g \circ h \in I_1(T'', S) \rightarrow T'' \Vdash (A \mapsto f \circ h)(g \circ h)),$$

and it is not hard to see that this is equivalent to the second formula of (iii).

(iv) Easy.

(v) Formula induction.

(vi) Easy.

(vii) (a) By 3.4.5 and the fact $S \vdash A = S \vdash (A \mapsto f)$.

(b) $T \Vdash \exists x \alpha x = \exists \phi \in I_0(T)(T \vdash A \phi)$; as $\phi$ is continuous, we have $\phi \circ (y^*)$ is constant, for some $y \in T$, so by 3.4.10 and 3.4.9(i), $\exists x (T \vdash A(\alpha_x - x))$; hence $\exists x (T \vdash \alpha x)$, by (a) and 3.4.9(ii).

(viii) Formula induction: use (vi). □

3.4.13. **Theorem (Soundness)**. $T_1 \vdash A \Rightarrow T_2 \vdash \forall T \vdash (T \vdash A)$.

**Proof.** Induction over the length of a proof of $A$.

*Logical axioms and rules of TAPP:*

$A \rightarrow A$, $\forall x \alpha x \rightarrow A \tau$: trivial, for $\tau$ contains no choice variables.

$A \tau \rightarrow \exists x \alpha x$: use 3.4.9(ii) and $I_0 \text{AX1}$.

\[
\frac{A}{B \rightarrow A} : \text{trivial, by 3.4.5.}
\]

\[
\frac{A \rightarrow B}{R \rightarrow C} : \text{easy, by 3.3.11(ii).}
\]

\[
\frac{A \rightarrow B}{A \rightarrow A} : \text{easy, by 3.3.8(ii).}
\]

\[
\frac{A \rightarrow B}{A \rightarrow C} : \text{trivial.}
\]

\[
\frac{(A \land B) \rightarrow C}{A \rightarrow (B \rightarrow C)} : \text{assume } T \vdash A \land B \rightarrow C, \text{ i.e.}
\]

\[
(6) \quad \forall S \forall f \in I_1(S, T)(S \vdash A \mapsto f \land S \vdash B \mapsto f \rightarrow S \vdash C \mapsto f).
\]

This implies

$$\forall S \forall f \in I_1(S, T) \forall S' \forall g \in I_1(S', S)(S \vdash A \mapsto f \circ g \land S \vdash B \mapsto f \circ g \rightarrow S \vdash C \mapsto f \circ g).$$
Distribute $\forall S, \forall g \in I_1(S', S)$:

$$\forall S \forall f \in I_1(S, T)(\forall S' \forall g \in I_1(S', S) S' \models A \uparrow f \circ g$$
$$\rightarrow \forall S' \forall g \in I_1(S', S)(S' \models B \uparrow f \circ g \rightarrow S' \models C \uparrow f \circ g)).$$

With 3.4.10:

$$(7) \quad \forall S \forall f \in I_1(S, T)(S \models A \uparrow f \rightarrow \forall S' \forall g \in I_1(S', S) (S' \models B \uparrow f \circ g$$
$$\rightarrow S' \models C \uparrow f \circ g)),$$

i.e., $T \models A \rightarrow (B \rightarrow C)$.

The other way round is easier: take $S' := S, g := \lambda x . x$ in (7) and we get (6).

$$\frac{A \rightarrow B}{A \rightarrow \forall x B} : \text{trivial.}$$
$$\frac{A \rightarrow B}{\exists x A \rightarrow B} : \text{assume } T \models A \rightarrow B, \text{ i.e.}$$

$$(8) \quad \forall S \forall f \in I_1(S, T)(S \models (A \uparrow f)(x) \rightarrow S \models B \uparrow f).$$

Let $f \in I_1(S, T)$ and assume

$$S \models (A \uparrow f)(\phi) \quad \text{for some } \phi \in I_0(S).$$

By 3.3.10(i), $\delta \mod \phi$ for some $\delta \in M_0(S)$. Now, by 3.4.11:

$$\forall a \in S \delta(A(\delta a))^\models (A \uparrow f \circ (\delta a*))((\phi \circ (\delta a*))).$$

Since $\delta \mod \phi$, we have

$$\forall a \in S \exists x \forall b \in S \delta(\delta a^0) \phi(\delta a^0 \ast b) = x,$$

so, with 3.4.9(ii)

$$\forall a \in S \exists x \forall b \in S \delta(\delta a^0)^\models (A \uparrow f \circ (\delta a*))(x).$$

With (8) this gives

$$\forall a \in S (S \delta(\delta a^0 \models B \uparrow (f \circ (\delta a*)))$$

which implies (by 3.4.11) $S \models B \uparrow f$.

So we have shown

$$\forall S \forall f \in I_1(S, T)(\exists \phi (S \models (A \uparrow f)(\phi)) \rightarrow S \models B \uparrow f),$$

i.e., $T \models (\exists x A \rightarrow B)$.

**Non-logical axioms of TAPP**: Most of them present no problems. We only consider IND: assume

$$T \models A 0 \uparrow f \quad \text{and} \quad T \models \forall n (A n \rightarrow A(n + 1)) \uparrow f,$$

i.e.

$$\forall n \forall S \forall g \in I_1(S, T)(S \models A n \uparrow f \circ g \rightarrow S \models A(n + 1) \uparrow f \circ g);$$

then $\forall n (T \models A n \uparrow f \rightarrow T \models A(n + 1) \uparrow f)$, so with $T \models A 0 \uparrow f$ we get $\forall n (T \models A n)$. 
Extended bar induction

Axioms and rules of TAPP* for sequence variables:
\( \forall \text{R}_\text{SEQ} \): let \( A = A(\bar{\beta}) \), \( B = B(\alpha, \bar{\beta}) \). Now \( T_2 \vdash \forall \mathcal{T}(T \vdash (A \rightarrow B)) \) reads
\[
T_2 \vdash \forall \mathcal{T} (\tilde{f}, g \in I_1(S, T) \rightarrow \forall S' \forall h \in I_1(S', S)(S' \vdash A(\tilde{f} \uparrow h) \\
\rightarrow S' \vdash B(g \circ h, \tilde{f} \uparrow h)));
\]
we quantify over \( S, \tilde{f}, g \), take \( S' := S, T := U, h := \lambda x . x \) and get
\[
T_2 \vdash \forall S \forall \tilde{f}, g \in I_1(S, U)(S \vdash A(\tilde{f}) \rightarrow S \vdash B(g, \tilde{f}));
\]
now take \( \tilde{f} := \tilde{f} \circ h \circ k \) and use 3.3.11(ii) and \( I_1(S, T) \subseteq I_1(S, U) \) (by 3.3.4(vi)):
\[
T_2 \vdash \forall S'' \forall T (\tilde{f} \in I_1(S'', T) \rightarrow \forall S' \forall h \in I_1(S', S'')(S'' \vdash A(\tilde{f} \circ h) \\
\rightarrow \forall S' \forall k \in I_1(S, S')(S' \vdash B(g, \tilde{f} \circ h \circ k)));
\]
Distribute \( \forall S, \forall k \in I_1(S, S') \) and apply 3.4.5:
\[
T_2 \vdash \forall T (T \vdash (A \rightarrow \forall \alpha B));
\]
\( \exists \text{R}_\text{SEQ} \): as above, but simpler: write out \( U \vdash (A \rightarrow \forall \alpha B) \) and use \( I_1(S, T) \subseteq I_1(S, U) \).
\( \forall \alpha \forall \alpha \rightarrow A \beta \): let \( A = A(\alpha, \bar{\gamma}) \). Now \( T \vdash (\forall \alpha A \alpha \rightarrow A \beta) \) reads
\[
g, \tilde{h} \in I_1(S, T) \rightarrow \forall S' \forall k \in I_1(S', S)(S' \vdash A(g) \\
\rightarrow \forall f \in I_1(S'', U)(S'' \vdash A(f, \tilde{h} \circ k)))
\]
and this holds in \( T_2 \): to see this, take \( S'' := S, I := \lambda x . x, f := g \circ k \) and use \( I_1(S, T) \subseteq I_1(S, U) \).
\( A \beta \rightarrow \exists \alpha A \alpha \): Let \( A = A(\alpha, \bar{\gamma}) \). Now \( T \vdash (A \beta \rightarrow \exists \alpha A \alpha) \) reads
\[
g, \tilde{h} \in I_1(S, T) \rightarrow \forall S' \forall k \in I_1(S', S)(S' \vdash A(g \circ k, \tilde{h} \circ k) \\
\rightarrow \exists f \in I_1(S', U)(S' \vdash A(f, \tilde{h} \circ k)))
\]
which evidently holds (take \( f := g \circ k \)).

SEQAX1: \( T \vdash \forall \alpha \forall \forall \exists \alpha (an = x) \) reads (see 3.4.7(i))
\[
\forall S \forall g \in I_1(S, U) \forall n \exists \phi \in I_0(S) \forall a \in \bar{S} (gan = \phi a)
\]
and this holds by the definition of \( I_1(S, U) \).

SEQAX2: \( T \vdash \forall x \exists \alpha \forall \forall (xn = an) \) reads (3.4.7(ii))
\[
\forall x \exists g \in I_1(T, U) \forall n \forall a \in \bar{S} (xn = gan)
\]
and this is a consequence of I0AX1 and the definition of \( I_1 \).

SEQAX3: \( T \vdash \forall \alpha \beta \exists \gamma \forall \forall (\gamma n = (an, \beta n)) \) reads
\[
\forall S \forall f \in I_1(S, U) \forall S' \forall g \in I_1(S', S) \forall h \in I_1(S', U) \\
\exists k \in I_1(S, U) \forall n \forall a \in \bar{S}' (kan = (f(ga)n, han))
\]
and this follows from 3.3.8(vi) and 3.3.11(ii).
SEQAX4: $T \vDash \forall x \exists \beta (\beta \circ x = \lambda \forall n (\beta (n + 1) = \alpha n))$ reads

$$\forall S \forall f \in I_1(S, U) \forall x \exists g \in I_1(S, U) \forall a \in S (\gamma a 0 = x)$$

$$\land \forall n \forall a \in S (\gamma a (n + 1) = \gamma a n)$$

and this follows from the definition of $I_1(S, U)$.

Tree axioms and rules of $T^*_1$: 

$\forall R_{TR}, \exists R_{TR}, \forall A_{TR}, \exists A_{TR}$: easy, since $\forall T$, $\exists T$ commute with $\vdash$.

TRAX1–8: also easy, for they do not contain sequence variables.

$I_0AX1$: $S \vdash (\forall \alpha \in \beta (\phi \alpha = x) \rightarrow \phi \in I_0(\tau))$ reads

$$\forall S' (\forall S'' \forall g \in I_1(S'', T) \forall a \in S'' (\phi (ga) = x) \rightarrow \phi \in I_0(\tau))$$

and this holds in $T_2$ (take $S'' := T, g := \lambda x \cdot x$).

$I_0AX2$: easy, as it contains no sequence variables.

$I_0AX3$: using $(\exists x A \rightarrow B) \leftrightarrow \exists x (A \rightarrow B)$ and $(A \lor B \rightarrow C) \leftrightarrow ((A \rightarrow C) \land (B \rightarrow C))$, we can rewrite $I_0AX3$ without $\lor$ and $\exists$. Now the proof of $T \vdash I_0AX3$ is analogous to that for $T \vdash I_0AX1$.

$I_1AX$: $T \vdash \forall \alpha \in S_1 \forall f \in I_1(S_1, S_2) \exists \beta \in S_2 \forall n (\beta n = \gamma a n)$ reads

$$\forall T' \forall g \in I_1(T', S_1) \forall f \in I_1(S_1, S_2) \exists h \in I_1(T', S_2)$$

$$\forall a \in \beta' \forall n (\gamma a n = f (ga n))$$

and this follows from $3.3.11(ii)$.

$ECS1$: $S \vdash (\forall a \in T A a \rightarrow \forall \alpha \in T A a)$ reads (remember that $A$ is prime)

$$\forall S' (\forall a \in T A a \rightarrow \forall S'' \forall f \in I_1(S'', T) \forall b \in S'' A (fb))$$

and this follows from the definition of $I_1$.

$ECS2$: Both $S \vdash (\forall \alpha (A a \rightarrow B \alpha))$ and $S \vdash (\forall T \forall f \in I_1(T, U) (\forall \alpha \in T A (f \alpha) \rightarrow \forall \alpha \in T B (f \alpha)))$ are equivalent to

$$\forall T \forall f \in I_1(T, U) (T \vdash A f \rightarrow T \vdash B f);$$

use $3.4.7(iv)$ for the second equivalence.

$ECS3$: $S \vdash \forall \alpha \in T \exists A A \alpha, x)$ and $S \vdash \exists \phi \in I_0(T) \forall \alpha \in T A (\alpha, \phi \alpha)$ are equivalent to

$$\exists \phi \in I_0(T) (T \vdash A (\lambda x \cdot x, \phi))$$

$ECS4$: analogous to $ECS3$.

$EAC$: easy, by $3.4.12(vii)$ (recall that $EAC$ does not contain free sequence variables). □

We complete the picture of $T^*_1, T_2$ and $\vdash$ as follows.

**3.4.14. Theorem.** (i) Let $A$ be a completely regular formula of $T^*_1$. Then

$$T^*_1 \vdash A \leftrightarrow (T \vdash A).$$
(ii) Let $A$ be a formula of $T_2$. Then

$$T_2 \vdash A \leftrightarrow (T \models A).$$

**Proof.** (i) With formula induction we show, for completely regular $A$:

$$T_1^* \vdash \forall \alpha \in \tilde{T} A(\tilde{p}\alpha) \leftrightarrow T \models A(\tilde{p});$$

from this (i) follows.

A prime: by ECS1.

$A = B \wedge C$, $A = \forall x B x$: easy.

$A = B \rightarrow C$: simple, use ECS2.

$A = \forall \beta B\beta$: by the definition of $\models$ and the induction hypothesis we see that

$$T \models \forall \beta B(\tilde{p}, \beta)$$

is equivalent to

$$\forall S \forall f \in I_1(S, T) \forall g \in I_1(S, U) \forall \alpha \in \tilde{S} B(\tilde{p}(f\alpha), g\alpha);$$

now (1) $\leftrightarrow \forall \alpha \in \tilde{T} \forall \beta B(\tilde{p}\alpha, \beta)$: is evident; for $\rightarrow$, take $S := T \times U$, $f := \pi_0$, $g := \pi_1$ and use substitution for $= (A$ is regular, hence $B)$.

$A = \exists \beta B\beta$: use ECS4 and the induction hypothesis.

$A = \exists x B x$: analogous.

(ii) We prove with formula induction:

$$T_2 \vdash A(\tilde{x}) \leftrightarrow T \models A(\tilde{p}),$$

here the $\tilde{p}$ are constant parameters with value $\tilde{x}$, i.e., $\forall a \tilde{p}a = \tilde{x}$. From this (ii) follows.

A prime, $A = B \wedge C$, $A = \exists x B x$: easy.

$A = \exists y B y$: now $T \models \exists y B(y, \tilde{p}) = \exists \phi \in I_0(T)(T \models B(\phi, \tilde{p}))$; by the induction hypothesis and 3.3.10(i) this is equivalent to

$$\exists \delta \in M_0(T) \exists \phi \in I_0(T)(\delta \text{ mod } \phi \wedge \forall a \in \tilde{T} (T_{a(\delta a)} \models B(\tilde{p} \upharpoonright (a(\delta a) \star), \phi \circ (a(\delta a) \star))));$$

i.e. (by 3.4.4(i)),

$$\exists \delta \in M_0(T) \exists \phi \in I_0(T)(\delta \text{ mod } \phi \wedge \forall a \in \tilde{T} \exists y (T_{a(\delta a)} \models B(\tilde{p} \upharpoonright (a(\delta a) \star), \lambda z \cdot y))).$$

With the induction hypothesis:

$$\exists \delta \in M_0(T) \exists \phi \in I_0(T)(\delta \text{ mod } \phi \wedge \forall a \in \tilde{T} \exists y B(\tilde{x}, y)),$$

i.e., $\exists y B(\tilde{x}, y)$. □

3.5. **Reduction to arithmetic**

In this section the proof of our main theorem is completed.
3.5.1. First we define a new theory $T_3$ which looks like $T_2$, but without tree variables. Let $I_0AX'1'–3'$ be the following axiom schemata ($A$ an arbitrary negative formula of TAPP):

$I_0AX'1'$ Tree($A$) $\land \forall a \in A (\phi a = x) \rightarrow \phi \in I_0(A)$

$I_0AX'2'$ Tree($A$) $\land \forall c \in A (\phi c \in I_0(A_c)) \rightarrow \phi \in I_0(A)$

$I_0AX'3'$ Tree($A$) $\land \forall x \in A \forall \phi [\exists y \forall a \in A (\phi a = y) \rightarrow (\forall \tilde{y} \in A (\phi \tilde{y} \in P(x \ast \tilde{y})) \rightarrow \phi \in P(x))]$

Now $T_3 := TAPP + I_0AX'1'–3' + EAC$.

3.5.2. Theorem. $T_2 \vdash A \Rightarrow T_3 \vdash A$ for $A \in \mathcal{L}(T_3)$.

Proof. A detailed proof would be long and tedious, so we confine ourselves to a sketch. Let $T_2^*$ be an arbitrary subtheory of $T_2$ with only finitely many instances of TRAXS, say for the formulae $A_1, \ldots, A_n$. We assume $\text{FV}(A_i) \subseteq \{x, z_i\}, i = 1, \ldots, n$ (the variable $x$ is used to define the set $A_i$; see 3.1.9). For technical reasons, we add $A_0 := (\text{tth } x \in \mathbb{N})$ to the list $A_1, \ldots, A_n$. We shall define an interpretation $\mathcal{I} : T_2 \rightarrow T_3$ satisfying

$T_2 \vdash A \Rightarrow T_3 \vdash A_1;$

from this the theorem follows.

The naive idea for $\mathcal{I}$ is: replace formulae $\forall TA [\tau_j \in T]$ by

$\hat{\mathcal{I}} \bigwedge_{j=0}^\infty (\text{Tree}(A_i) \rightarrow A_i(\tau_j))$.

But this is not enough, for the $A_i$ may contain parameters, and we also have to deal with the closure conditions $\forall T \forall x (x \in T \rightarrow \exists S (S = T \& T'))$ (TRAX6) and $\forall TT' \exists S (S = T \times T')$ (TRAX7). This leads us to considering the “universe of trees” of $T_2^*$, which consists of the trees defined by $A_0, \ldots, A_n$, closed off under taking subtrees and products.

We recall the notation $\times, [\_], [\_]$ from 3.2.1 and define the following notation:

$x^{(1)} := x,$

$x^{(0)} := [x^{(1)}];

$y = (y_0, \ldots, y_n), y_i = 0$ or $1$ ($i = 0, \ldots, n$). Such a sequence $y$ is called a $0$-$1$-sequence and we call $x^{(y)}$ the $y$-projection of $x$.

An example:

$x^{(1,0,0)} = [[[x]]]_0.$

We now have, e.g.,

$x \in (T_1 \times T_2) \times T_3 \leftrightarrow x^{(0,0)} \in T_1 \land x^{(0,1)} \in T_2 \land x^{(1)} \in T_3.$
The idea now is to code the trees of the "universe of trees" of $T_2^i$ by quintuples $y, z, u, v, m$ which satisfy

(i) $z, u, v$ are finite sequences with length $m$;
(ii) $z$ is a sequence of parameters;
(iii) $u$ is a finite sequence of different finite 0–1-sequences;
(iv) $v$ is a sequence of natural numbers $\leq n$;
(v) $\langle \rangle, z, u, v, m$ code a tree which contains $y$.

(i)–(v) are collected in $\text{Adm}(y, z, u, v, m)$:

$$\text{Adm}(y, z, u, v, m) := \text{lth } z = \text{lth } u = \text{lth } v = m \land m \in \mathbb{N} \land \forall i < m \left( (\text{lth}(u)_i \in \mathbb{N} \land (v)_i \in \mathbb{N} \right) \land \forall ij < m ((u)_i = (u)_j \rightarrow i = j) \land \forall i < m \forall k < \text{lth}(u)_i ( ((u)_i)_k \in \{0, 1\} \land \text{Tree}(T(\langle \rangle, x, z, u, v, m)) \land T(\langle \rangle, y, z, u, v, m),$$

where

$$T(\langle \rangle, y, x, z, u, v, m) := \forall i < m \left( \left( \mathop{\bigwedge}_{j=0}^i (j = (v)_j \rightarrow (y \ast x)^{(u)_j} \in A[z := (z)_i]) \right) \right).$$

We call $\{x \mid T(\langle \rangle, x, z, u, v, m)\}$ the tree coded by $y, z, u, v, m$; it consists of those $x$ for which holds:

for any $i < m$, the $(u)_i$-projection of $y \ast x$ is in the tree defined by the formula $A_{(v)_i}$ with parameters $(z)_i$.

Now the definition of $^\dagger$ is as follows.

$$\begin{align*}
(\forall T B)^\dagger &= \forall y z u v m \left( \text{Adm}(y, z, u, v, m) \rightarrow (B[T := T(\langle \rangle, y, x, z, u, v, m)])^\dagger \right) \\
(\exists T B)^\dagger &= \exists y z u v m \left( \text{Adm}(y, z, u, v, m) \land (B[T := T(\langle \rangle, y, x, z, u, v, m)])^\dagger \right)
\end{align*}$$

$^\dagger$ commutes with $\forall x, \exists y, \land, \lor, \rightarrow$ and leaves prime formulae unchanged.

By this definition of $^\dagger$, we get formulae like $\tau \in (T(\langle \rangle, y, x, z, u, v, m))_\sigma$ and $\tau \in (T(\langle \rangle, y, x, z, u, v, m)) \times T(\langle \rangle, y, x, z, u, v, m)$; to interpret these we recall the conventions

$$\tau \in A := A[x := \tau], \quad \tau \in A, := \sigma \ast \tau \in A$$

from 2.4, and adopt the following:

$$\begin{align*}
\tau \in U &:= \text{lth } \tau \in \mathbb{N}, \\
\tau \in A \times B &:= [\tau]_0 \in A \land [\tau]_1 \in B.
\end{align*}$$

We check the soundness of $^\dagger$ in the version

$$T_2^i \vdash A \Rightarrow T_3^i \vdash (\forall T A)^\dagger,$$
where $T$ are the free tree variables of $A$. By the definition of $f$, we only have to inspect the rules and axioms concerning trees, and EAC.

$\forall R_{TR}, \exists R_{TR}, \forall A_{TR}, \exists A_{TR}$: easy.

TRAX1: ($\forall T (\text{Tree}(T)))^f$ follows from the definition of Adm.

TRAX2–4: trivial, by the conventions mentioned above.

TRAX5: we only have instances with $A_i$, $1 \leq i \leq n$. Now

$$A_i(x, z) \iff T(\langle i \rangle, x, z, \langle i \rangle, i, 1)$$

and, by Tree($A_i(x, z)$), we have Adm($\langle i \rangle, z, \langle i \rangle, i, 1$).

TRAX6: if $T$ is coded by $y, z, u, v, m$, then take $y * x, z, u, v, m$ as code for $S (\equiv T \times T')$.

TRAX7: if $T$ is coded by $y, z, u, v, m$ and $T'$ by $y', z', u', v', m'$, then take $y \times y', z \pi z', (u)_0^*0, \ldots, (u')_0^*1, (u')_1^*1, \ldots, (u')_{m-1}^*1, v \pi v', m + m'$ as a code for $S (\equiv T \times T')$.

TRAX8: easy, for in $T_3$ we have

$$\text{Trcc}(A) \land \text{Trcc}(B) \land A = B \rightarrow I_0(A) = I_0(B)$$

for $A \in \mathcal{L}^{-}(\text{APP})$ (to be proved with induction over $I_0(A), I_0(B)$), and also $\text{Tree}(A) \rightarrow A \equiv A_0$.

EAC: it is enough to show that $A^f$ is negative if $A$ is. By the definition of $f$ we only have to check that $T(y, x, z, u, v, m)$ is negative, and this follows from the fact that the $A_i$ are negative (by the restriction on TRAX5).

This ends the proof. \(\square\)

Now we compare $T_3$ with $\text{TAPP(ID)} + \text{EAC}$.

3.5.3. Lemma. $T_3 \vdash A \Rightarrow \text{TAPP(ID)} + \text{EAC} \vdash A$ for $A \in \mathcal{L}(\text{TAPP})$.

Proof. We shall show that $I_0\text{AX1}′-3′$ hold in $\text{TAPP(ID)}$. Let $B_A = B_A(P, z)$ be defined by (we write $\langle x, \phi \rangle$ for $z$):

$$B_A(P, \langle x, \phi \rangle) = [\exists y \forall a \forall n (x * a n e A) \rightarrow \phi a = y) \land \forall y (x * \hat{y} e A \rightarrow \langle x * \hat{y}, \phi_y \rangle e P)] \rightarrow \langle x, \phi \rangle e P.$$ 

$\Gamma_{B_A}$ is the predicate operator with

$$z \in \Gamma_{B_A}(P) \iff B_A(P, z).$$

We write $I_A$ for $I_\Gamma (\Gamma$ abbreviates $\Gamma_{B_A}$), the least fixed point of $\Gamma_{B_A}$; in $\text{TAPP(ID)}$ we have

1. $\Gamma(I_A) \subseteq I_A$,
2. $\Gamma(P) \subseteq P \rightarrow I_A \subseteq P$.

Now we define $I_0(A_x)$ explicitly by

$$\phi \in I_0(A_x) : = \langle x, \phi \rangle \in I_A;$$
writing out (1), (2) and substituting $\phi \in I_0(A_\phi)$ for $\langle x, \phi \rangle \in I_A$ and $\phi \in P(x)$ for $\langle x, \phi \rangle \in P$ yields $I_0AX1\neg3\neg3$, even without the condition Tree($A$). □

3.5.4. **Theorem.** $TAPP^* + EBI_0 A \Rightarrow ID_1 \vdash A$ for $A \in \mathcal{L}(HA)$.

**Proof.** Let $TAPP^* + EBI_0 A$, $A \in \mathcal{L}(HA)$. Then, by 3.2.11:

$$T_1^* \vdash A.$$ 

By 3.4.13 and 3.4.14(ii) ($A$ is a fortiori in the language of $T_2$):

$$T_2^* \vdash A.$$ 

By 3.5.2 and 3.5.3:

$$TAPP(ID) + EAC \vdash A.$$ 

Finally, by 2.4.10:

$$ID_1 \vdash A.$$ □

To establish that $ID_1$ axiomatizes the arithmetical fragment of $TAPP^* + EBI_0$, we prove the converse of the previous theorem. We shall use a result by Sieg, for which we first need a definition.

3.5.5. **Definition.** Let $\{\cdot\}(\cdot)$ be the Kleene-bracket-notation; without loss of generality we may assume that $\forall n \{0\}(n) = 0$. We define the axioms $\mathcal{O}AX1\neg3$:

- $\mathcal{O}AX1 \quad 0 \in \mathcal{O}$
- $\mathcal{O}AX2 \quad \forall n \langle x \rangle(n) \downarrow \wedge \langle x \rangle(n) \in \mathcal{O} \rightarrow x \in \mathcal{O}$
- $\mathcal{O}AX3 \quad A(0) \wedge \forall x [\forall n \langle x \rangle(n) \downarrow \wedge A(\langle x \rangle(n)) \rightarrow Ax] \rightarrow \forall x \in \mathcal{O} Ax.$

$\mathcal{O}$ is called the inductively defined tree class of the first order. We also put

$$ID_1(\mathcal{O}) := \mathcal{O}AX1 + \mathcal{O}AX2 + \mathcal{O}AX3,$$

$$ID_1(\mathcal{O}) := HA + ID_1(\mathcal{O}).$$

3.5.6. **Theorem** (Sieg). $ID_1$ and $ID_1(\mathcal{O})$ prove the same arithmetical theorems.

**Proof.** Follows from [4, Ch. III, Theorem 3.2.3]. □

3.5.7. **Lemma.** $ID_1(\mathcal{O}) \vdash A \Rightarrow TAPP^* + BI \vdash A$.

**Proof.** We interpret $x \in \mathcal{O}$ by

1. $\forall \alpha \in \mathbb{N}^\omega \exists n (fx(\check{\alpha}n) = 0 \wedge \forall m < n (fx(\check{\alpha}m) > 0))$

where $f$ is the function satisfying

$$fx(\check{\cdot}) = x, \quad fx(y \star z) = \{fxy\}(z).$$
We verify that $\mathcal{O}A_1$–$3$ become derivable in $\text{TAPP}^* + \text{BI}$ under this interpretation. $\mathcal{O}A_1$ and $\mathcal{O}A_2$ follow, without using BI, by writing out their interpretation and using the definition of $f$; for $\mathcal{O}A_3$ we do need BI. Assume

(i) $A_0$,
(ii) $\forall x (\forall n A((x)(n)) \rightarrow A(x))$,
(iii) $x \in \mathcal{O}$, i.e. $\forall \alpha \in \mathbb{N} \exists n (fx(\tilde{a}n) = 0 \land \forall m < n (fx(\tilde{a}m) > 0))$.

Put

$$\text{By} := \forall z \in \mathbb{N}^\omega A(fx(y * z)).$$

Then

(a) $\text{Bar}(\mathbb{N}^\omega, B)$, by (i), (ii) and $\forall n (\{0\}(n) = 0)$;
(b) $\text{Mon}(\mathbb{N}^\omega, B)$, by the definition of $B$;
(c) $\text{Ind}(\mathbb{N}^\omega, B)$, by (ii) and the definition of $f$;
(d) $\text{Tree}(\mathbb{N}^\omega)$.

So with BI we get $B(\langle \rangle)$, hence $A(fx(\langle \rangle))$, i.e., $Ax$. We conclude $\forall x \in \mathcal{O}A_x$, so $\mathcal{O}A_3$ is derived. $\Box$

3.5.8. Theorem. $\text{TAPP}^* + \text{EBI}_0 \vdash A \iff \text{ID}_1 \vdash A$ for $A \in \mathcal{L}(HA)$.

Proof. Combine 3.5.4 and 3.5.7. $\Box$

We now formulate the principal corollary. Let $\text{EL}^*$ be the theory $\text{EL}$ but with $\alpha, \beta, \ldots$ as sequence variables. In $\text{EL}^*$ we can write down $\text{Tree}(A)$, $\text{Bar}(A, P)$, $\text{Mon}(A, P)$, $\text{Ind}(A, P)$ and $\text{EBI}_0(A, P)$ just as in $\text{TAPP}^*$ (now $x, y, \alpha$ range over natural numbers); $\text{EBI}_0$ for $\text{EL}^*$ is defined as $\text{EBI}_0(A, P)$ for all $P \in \mathcal{L}(\text{EL}^*)$ and all $A \in \mathcal{L}(HA)$.

3.5.9. Theorem. $\text{EL}^* + \text{EBI}_0$ and $\text{ID}_1$ are arithmetically equivalent.

Proof. $\text{EL}^* + \text{EBI}_0$ can be embedded into $\text{TAPP}^* + \text{EBI}_0$ by extending the natural embedding of $\text{HA}$ into $\text{TAPP}$ (see 2.1.1) with

$$\forall \alpha A \mapsto \forall \alpha (\forall n (an \in \mathbb{N}) \rightarrow A), \quad \exists \alpha A \mapsto \exists \alpha (\forall n (an \in \mathbb{N}) \land A).$$

This takes care of one direction; the other is established by transferring the proof of 3.5.7. to $\text{EL}^* + \text{BI}$. $\Box$

A. Appendix (by A.S. Troelstra)

A.1. Due to an oversight, the proof of the elimination theorem for $\text{CS}^*$ in [28] is defective. A correction of the proof by modifying the definition of the elimination translation results in weaker versions of some of the principal results of [28]. The contents of sections not involving the elimination theorem are not affected; thus no changes are needed in Sections 1.1–1.5, 1.8, 1.9, 1.11, 2, 3, 6.2, 6.3, 6.5.
A.2. Let us first indicate where the difficulty in the proof of the elimination theorem in [28] originates. From recent work (e.g. [15]) it has become clear that the elimination translations for various well-known theories of choice sequences, such as the theory LS of lawless sequences and CS, the theory discussed in [20] and [15], can be reformulated in terms of forcing over a site (forcing over a category provided with a Grothendieck topology). Now the elimination translation as defined in [28] does not in any obvious way correspond to a site; the analytic-data type axiom (1) does not mesh naturally with the other axioms, and accordingly the implication-clause of the elimination translation does not agree with the other cases. An intuitively correct version of axiom (1) would be

(1) \( \forall \alpha^\sigma (A \alpha \to B \alpha) \to \forall \tau \in T \forall e^\tau \alpha (\forall e^\tau \alpha (e \mid \alpha) \to \forall e^\tau \alpha B(e \mid \alpha)) \),

where \( T \) is a collection of tree predicates closed under the formation of products and restriction to subtrees determined by initial segments. This is essentially also the form of "analytic data" of the elimination translation or forcing notion of this paper (3.4).

If type \( \sigma \) represents a smaller tree than \( \tau \), then in CS* as formulated in [28] the range of the \( \alpha^\sigma \) is supposed to be contained in the range of the \( \beta^\tau \). More precisely, if \( \sigma = [A] \), \( \tau = [B] \), \( \forall x (Ax \to Bx) \), then by (IV) and (V) in [28, 4.2] we have \( \forall \alpha^\sigma \exists \beta^\tau (\alpha^\sigma = \beta^\tau) \). This becomes problematic in combination with (1) of [20, 4.2]. For on the one hand, (1) "explains" \( \forall \alpha^\sigma (A \alpha \to B \alpha) \) as

(2) \( \forall e^\alpha \sigma (\forall e^\alpha \sigma A(e \mid \alpha) \to \forall \alpha^\sigma B(e \mid \alpha)) \).

On the other hand we must require that adding quantifiers over dummy variables results in provably equivalent statements, and so (\( \beta \) not free in \( A, B \))

(3) \( \forall \alpha^\sigma (A \alpha \to B \alpha) \leftrightarrow \forall \alpha^\sigma \beta^\tau (A \alpha \to B \alpha) \).

Application of (1) to the right-hand side yields

(4) \( \forall \alpha^\sigma (A \alpha \to B \alpha) \leftrightarrow \forall \gamma^{\alpha \times \tau} (A(j, \gamma) \to B(j, \gamma)) \)

\( \leftrightarrow \forall e^{\alpha \times \tau} \gamma^{\alpha \times \tau} (\forall e^{\alpha \times \tau} A(j, e \mid \gamma) \to \forall \gamma^{\alpha \times \tau} B(j, e \mid \gamma)) \).

As a rule, the last formula in (4) will be stronger than (2); thus axiom (1) does not determine the meaning of \( \forall \alpha^\sigma (A \alpha \to B \alpha) \) unambiguously. Consequently, the proof of the elimination theorem for CS* already breaks down for the rules of propositional logic. The proof in [28] relies for many details on the corresponding proof in [20], where the problem does not arise because of the existence of a primitive recursive bijection between \( \mathbb{N} \times \mathbb{N} \) and \( \mathbb{N} \).

A.3. Replacing (1) by its modification (1) does not produce our original proof-theoretical results however, since the class of all tree predicates definable in EL is not itself explicitly definable in EL or CS*. However, if we consider EL with only finitely many instances EBl_n(A_i) (\( i \leq n \)) of extended bar induction of type 0 with \( A_i \) almost negative and arithmetical, then we can embed this system
into a system $CS'$ say, which permits elimination of choice sequences. In $CS'$ we have to include appropriate continuity axioms, and an axiom of type (1) where $T$ is now a \textit{definable} class of tree predicates closed under products, subtrees determined by initial sequents, and containing the trees $A_i^T(i \leq n)$. The lawlike part $IDB^*_1$ to which the elimination translation reduces $CS'$, is similar to $IDB^*$ in [28], except that axioms $K_{[A]}1-3$ are assumed for almost negative arithmetical $A$ only. Afterwards we can lift the restriction to almost negative $A_i$ by a combination with numerical realizability as in [28], and one obtains instead of Result 1 in [28]:

\textbf{Theorem.} EL + $EBI^*_o$ is conservative over ID$_1$ with respect to arithmetical sentences in $\Gamma^a_0$. Here $EBI^*_o$ is $EBI_o$ restricted to arithmetical tree predicates.

The proof of the stronger result 3.5.9 above follows the same pattern, except that abstract realizability replaces numerical realizability, and that for the final result one needs a Goodman-type theorem showing conservativeness of forms of the axiom of choice w.r.t. arithmetical sentences.

The method shows at the same time that $CS'$, obtained by adding to $IDB^*_1$ the schema $EBI^*_o$ and in addition continuity axioms of type II [28, 4.2] for $\sigma = [A]$ arithmetical and almost negative, is conservative over ID$_1 \cap \Gamma^a_0$.

\textbf{A.4.} An extension of the theorem above to obtain a reduction of EL* + $EBI^*_o(A)$ for almost negative $A$ in the language of $IDB^*$, to $IDB^*$ is presumably straightforward. Presumably it is also possible to carry through an elimination theorem and hence a reduction to $IDB^*$ for a system $CS''$ containing $EBI_o$, where the ranges of choice variables for distinct (tree) types $\sigma$, $\tau$ are regarded as completely independent and disjoint. Theorems of this kind are not elegant, but can be said to “approximate” to some extent the statement we originally intended to prove, namely that the proof-theoretic strength of EL + $EBI_o$ is the same as that of the theory of finitely iterated inductive definitions, and lead us to conjecture that our original claim is nevertheless true.

\textbf{References}